| | |
|---|---|
| **Titre:** Title: | Méthodes sans factorisation pour la tomographie à rayons-X en coordonnées cylindriques |
| **Auteur:** Author: | Maxime McLaughlin |
| **Date:** | 2017 |
| **Type:** | Mémoire ou thèse / Dissertation or Thesis |
| **Référence:** Citation: | McLaughlin, M. (2017). Méthodes sans factorisation pour la tomographie à rayons-X en coordonnées cylindriques [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/2742/ |

UNIVERSITÉ DE MONTRÉAL

# MÉTHODES SANS FACTORISATION POUR LA TOMOGRAPHIE À RAYONS-X EN COORDONNÉES CYLINDRIQUES

MAXIME MCLAUGHLIN
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
AOÛT 2017

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

MÉTHODES SANS FACTORISATION POUR LA TOMOGRAPHIE À RAYONS-X EN COORDONNÉES CYLINDRIQUES

présenté par : MCLAUGHLIN Maxime
en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées
a été dûment accepté par le jury d'examen constitué de :

M. LE NY Jérôme, Ph. D., président
M. ORBAN Dominique, Doctorat, membre et directeur de recherche
M. GOUSSARD Yves, Doctorat, membre et codirecteur de recherche
M. DUSSAULT Jean-Pierre, Ph. D., membre externe

# REMERCIEMENTS

L'accomplissement de ma maîtrise et de cet ouvrage ne revient pas qu'à moi seul. Je désire profiter de cette opportunité afin de témoigner de ma gratitude à vous tous qui m'avez supporté de près ou de loin. Avant tout, je tiens à remercier les membres du jury qui évalueront ce travail : Mr. Jérôme Le Ny, qui a eu l'amabilité de présider ce jury, Mr. Jean-Pierre Dussault, membre externe, ainsi que mes codirecteurs Mr. Dominique Orban et Mr. Yves Goussard.

Pour toute l'aide qu'ils m'ont apportée au cours de ces derniers mois, je tiens à remercier les professeurs Dominique Orban et Yves Goussard. Plus particulièrement, je vous remercie, Mr. Orban, pour tous ces échanges au sujet de l'optimisation et votre patience face à mes maintes interrogations. Cela a définitivement contribué à alimenter ma curiosité et à motiver mon désir d'apprendre.

Pour m'avoir encouragé et motivé à l'approche de la complétion de mes travaux, je tiens à remercier mes collègues de bureau au GERAD. Ces longues heures de rédaction auraient été plus difficiles sans vous.

Finalement, pour leur amour et leur support éternel dans mes succès comme mes échecs, je tiens à remercier mes parents et ma copine. Tout cela n'aurait pas été possible sans vous. Merci à tous mes amis, de près ou de loin, qui ont su rendre agréables ces longues années d'études. Je vous remercie tous du fond du coeur.

# RÉSUMÉ

Cet ouvrage s'inscrit à la suite de travaux portant sur l'étude du développement de resténoses par l'entremise de la tomographie à rayons X, ce qui requiert la reconstruction d'images haute résolution. À cette fin, nous considérons un algorithme de reconstruction *itératif*, basé sur le maximum a posteriori, qui permet une modélisation plus *réaliste* du processus d'acquisition des données. Soulignons que notre modélisation produit un problème aux moindres carrés régularisé sous contraintes de bornes. Les méthodes de reconstruction itératives sont reconnues pour produire des images de qualité, au détriment d'une augmentation drastique du stockage mémoire nécessaire et du temps de calcul.

Afin de pallier la consommation mémoire prohibitive impartie par une discrétisation du sujet en coordonnées cartésiennes, nous utilisons une discrétisation en coordonnées cylindriques, qui permet d'exploiter la symétrie du processus d'acquisition des données. Celle-ci mène à un opérateur de projection bloc-circulant. Le mauvais conditionnement de cet opérateur est mitigé par la mise à l'échelle diagonale dans le domaine de Fourier, mais transforme les contraintes de bornes en inégalités linéaires.

Ce travail a donc pour but d'étudier les différents solveurs sans factorisation pouvant résoudre un problème aux moindres carrés régularisé sous contraintes d'inégalités linéaires et d'identifier la méthode la plus performante. Nous émettons l'hypothèse qu'il est possible de projeter efficacement dans l'ensemble d'inégalités linéaires en traitant le dual du problème de projection, qui correspond à un problème aux moindres carrés sous contraintes de bornes. Suivant cette hypothèse, nous nous concentrons sur les méthodes d'ensemble actif à base de projections, notamment le gradient projeté spectral, une méthode d'ordre un, et une variante de l'algorithme TRON, une méthode d'ordre deux. Pour ce qui a trait au problème de projection, nous considérons différentes saveurs de méthodes de Newton projetées pour problèmes bornés. Nos résultats démontrent que notre variante de TRON résout efficacement le problème de reconstruction mis à l'échelle en coordonnées cylindriques. Elle offre même une performance similaire à L-BFGS-B sur le problème (non mis à l'échelle) en coordonnées cartésiennes, tout en ayant une empreinte mémoire considérablement inférieure. Nous discutons également de l'importance de l'exactitude des projections pour des méthodes à base de projection inexactes, telle l'approche que nous avons considérée.

# ABSTRACT

This work was done in the context of previously established results relating to restenosis development via X-ray tomography, which calls for the reconstruction of high resolution images. To this effect, we consider an iterative reconstruction algorithm that relies on maximum a posteriori estimation and that describes the physical phenomena occuring during the data acquisition process more accurately. We stress that this procedure comes down to minimizing a regularized least squares problem under positivity constraints. Iterative reconstruction methods are known to produce high quality images at the expense of greater memory requirements and computational time.

In order to circumvent the prohibitive memory requirements that occur when the object is discretized under cartesian coordinates, we use cylindrical coordinates, which allows us to exploit symmetries in the data acquisition process. In our algorithm, those symmetries materialize as a block-circulant projection matrix. The poor conditioning of this operator is mitigated by a diagonal scaling in the Fourier domain, but transforms the bound constraints into linear inequalities.

Hence, our main objective is to study the various factorization-free solvers that can be applied to convex non-linear problems under linear inequalities and to identify the most efficient approach. Our work revolves around the hypothesis that efficient projections into the constraint set can be designed by considering the dual projection problem, which comes down to a bounded least squares problem. Consequently, we focus our attention on projection-based active-set methods, namely the spectral projected gradient and an adaptation of TRON, respectively first-order and second-order methods. For the projection problem, we consider different flavors of projected Newton method for bounded problems. Our results show that our variant of TRON efficently solves the rescaled reconstruction problem under cylindrical coordinates. It is even competitive with the likes of L-BFGS-B applied to the reconstruction problem under cartesian coordinates, while consuming drastically less memory. We also discuss the importance of the exactness — or accuracy — of the projections for inexact projection-based methods, such as those we considered.

# TABLE DES MATIÈRES

# LISTE DES TABLEAUX

# LISTE DES FIGURES

# LISTE DES ANNEXES

# CHAPITRE 1 INTRODUCTION

La tomographie est une modalité d'imagerie médicale ayant pour objectif de produire une image de l'intérieur d'un patient afin de diagnostiquer une pathologie (Prince et Links, 2007). L'utilisation d'une source de rayonnement, par exemple des rayons X, permet d'irradier un objet et une image peut être reconstruite en traitant les intensités transmises. Les caractéristiques d'un problème de tomographie par rayons X sont : un volume important de données à traiter, une géométrie d'acquisition complexe, l'interaction non localisée entre le rayonnement et l'objet sous étude ainsi que la nécessité de traiter les données brutes afin de pouvoir les interpréter. Cette tâche est effectuée par l'entremise d'un *algorithme de reconstruction*.

Ces algorithmes sont principalement divisés en deux familles, soit les méthodes dites *analytiques* et *itératives*. Où les premières offrent une représentation intuitive, voire naïve, des phénomènes physiques sous-jacents, les secondes permettent de modéliser le processus d'acquisition des données et de choisir l'estimateur, faisant en sorte qu'une meilleure résolution peut être atteinte (Giovannelli et Idier, 2013, chapitre 1). En contrepartie, cette modélisation plus réaliste vient à un coût computationnel considérablement plus élevé, de sorte que peu d'intérêt était accordé aux méthodes itératives dans le domaine de la tomographie par rayons X. Historiquement, les manufacturiers de tomographes ont préféré la simplicité des méthodes *analytiques*, en cherchant plutôt à pallier leurs inconvénients en améliorant les appareils (Pan *et al.*, 2009).

C'est avec l'amélioration continue de la performance des ordinateurs, ainsi que les récentes demandes de diminution de la dose utilisée lors des scans, que l'engouement autour des méthodes itératives s'est vu renouvelé. Dans ce document, nous présentons un processus de reconstruction itératif, basé sur une approche probabiliste bayésienne, qui a pour objectif d'approcher l'imagerie haute résolution en des temps raisonnables. L'objectif à long terme serait de développer une approche itérative qui soit applicable dans un cadre clinique.

## 1.1 Définitions et concepts de base

Cette section se veut un survol du fonctionnement de la tomographie par rayons X ainsi que des familles d'algorithmes de reconstruction analytiques et itératives. Nous dédions la section 1.1.2 à la loi de Beer-Lambert, qui modélise la transmission d'un faisceau énergétique dans un objet, car elle constitue le fondement de tout algorithme de reconstruction. À la section 1.1.4, nous illustrons de quelle façon un cadre probabiliste peut être adopté afin

d'obtenir un problème d'optimisation qui mène ultimement à la reconstruction d'une image.

### 1.1.1 Tomographie par rayons X en bref

La tomographie par rayons X est une modalité d'imagerie médicale utilisée pour produire une image bidimensionnelle ou tridimensionnelle de l'intérieur d'un patient, typiquement à des fins de diagnostique clinique. En mesurant l'*atténuation* de l'énergie du faisceau de rayons X transmis au travers du patient, les algorithmes de reconstruction produisent une estimation des coefficients d'atténuation linéaires qui composent le corps du patient (Prince et Links, 2007). C'est donc l'*opacité* des tissus biologiques aux rayons X, i.e. leur capacité à atténuer l'énergie des rayons X, qui permet de produire un contraste et donc une image. Pour ce faire, une source rayons X, qui irradie le patient, et un ensemble de détecteurs tournent à l'unisson (autour du patient) afin de produire une collection de mesures d'intensités atténuées. En pratique, le patient est couché sur une table qui subit une translation axiale, tandis que l'anneau, qui contient les détecteurs et la source, tourne dans le plan transversal, tel qu'illustré schématiquement à la figure 1.1. Cette procédure de translation et de rotation permet de créer une succession de *tranches*, résultant en une modélisation tridimensionnelle des coefficients d'atténuation des tissus qui constituent le corps du patient.

On peut se représenter ce processus intuitivement en considérant que la source émet un faisceau plat de rayons parallèles et que la rotation et la translation ont lieu successivement. Ce cadre, qui revient à traiter un problème bidimensionnel, est employé pour illustrer les méthodes analytiques à la section 1.1.3 et est propre à une des premières générations de tomographes. Pour plus d'information sur les différentes générations de tomographes, le lecteur peut se référer à, e.g., Prince et Links (2007) ou encore Herman (2009).

Les tomographes modernes, qui sont généralement utilisés pour de l'imagerie tridimensionnelle, utilisent des faisceaux *coniques* et un panneau muni de plusieurs barrettes de détecteurs, de sorte que plusieurs tranches sont imagées simultanément (Prince et Links, 2007). En conséquence, il est possible de déplacer la source et l'anneau de détecteurs selon une trajectoire *hélicoïdale*, accélérant d'autant plus le processus d'acquisition.

### 1.1.2 La loi de Beer-Lambert stochastique

En tomographie par rayons X, la modélisation de l'atténuation d'un faisceau d'énergie le long d'une trajectoire dans un objet est décrite par la loi de Beer-Lambert. Un objet, ou patient, est modélisé par la distribution de coefficients d'atténuation linéaires $\mu(x, E) : \mathbb{R}^{n_{\text{dim}}+1} \to \mathbb{R}$, avec $x \in \mathbb{R}^n$ une position spatiale et $E \in \mathbb{R}$ l'énergie de la source, où $n_{\text{dim}} \in \{1, 2, 3\}$ représente

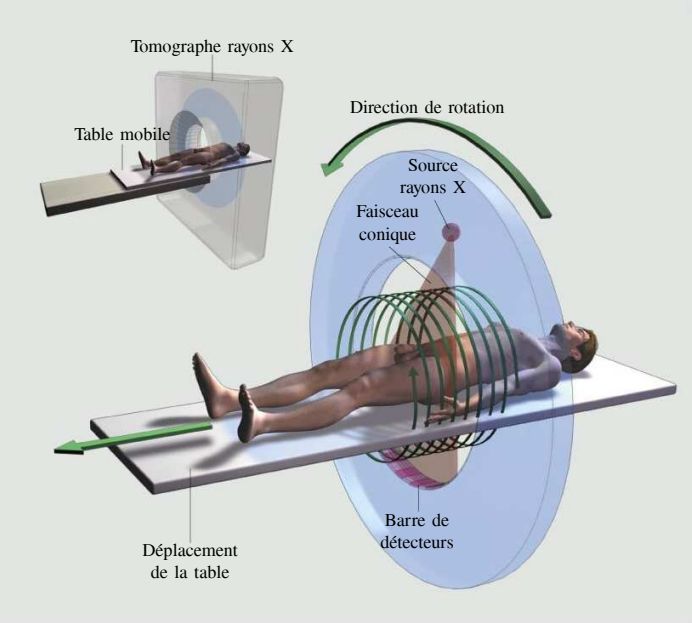



Figure 1.1 Principe de la tomographie, adapté de Brenner et Hall (2007)

la dimension de $x$. Supposons également que l'incertitude sur les mesures d'intensité aux détecteurs provient majoritairement des effets quantiques liés à l'atténuation des rayons X par les tissus. Sous ces hypothèses, Sauer et Bouman (1993) modélisent ces comptes de photons selon une distribution poissonnienne, notée $\mathcal{P}(l)$ [1]. Le paramètre $l$, i.e. la moyenne et la variance de la distribution, est l'atténuation prédite par la loi de Beer-Lambert *déterministe*. En conséquence, on introduit la variable aléatoire $N$, de réalisations $n \in \mathbb{R}^{n_{\text{meas}}}$ :

$$N \sim \mathcal{P}\left(\int_0^{E_{\max}} n_0(E)\, \mathrm{e}^{-\int_{L_i} \mu(x,E)\, \mathrm{d}\, x}\, \mathrm{d}\, E\right), \tag{1.1}$$

où $n_{\text{meas}}$ est le nombre de mesures, $n_0(E)$ représente l'intensité incidente de la source et $L_i$ est une trajectoire rectiligne. En pratique, $n_{\text{meas}}$ est fonction de la discrétisation angulaire, du nombre de détecteurs et du nombre de rangées de détecteurs, mais nous nous contentons d'introduire cette notation arbitrairement afin de faciliter la présentation. Nous utilisons l'indice $i = 1, \ldots, n_{\text{meas}}$ pour dénoter un $i^{\text{ème}}$ triplet angle-détecteur-position axiale, i.e. $L_i$ représente la droite reliant la source et un détecteur à un angle donné et à une hauteur donnée. Cette loi se nomme la loi de Beer-Lambert *stochastique polychromatique*, car on prend en compte la dépendance de $n_0$ et de $\mu$ à l'énergie $E$.

1. Le cadre polychromatique n'est pas présenté explicitement dans (Sauer et Bouman, 1993), mais il est trivial d'adapter l'équation afin de débuter la discussion dans un contexte plus général. Le lecteur peut se référer à, e.g., (Herman, 2009) ou (Prince et Links, 2007) pour plus d'informations sur la loi de Beer-Lambert polychromatique.

Une simplification relativement commune est le passage du cadre polychromatique au cadre *monochromatique* par l'élimination de la dépendance à l'énergie. Les effets de cette approximation se manifestent principalement lorsque des milieux avec des coefficients d'atténuation linéaires variant fortement selon l'énergie sont présents. Cela résulte en l'apparition d'artefacts dans les images reconstruites, typiquement en forme "d'étoile", rendant difficile l'interprétation des résultats. Dans le cadre de nos travaux, nous posons l'hypothèse monochromatique et référons le lecteur aux travaux d'Hamelin (2009) en ce qui concerne une modélisation polychromatique du phénomène.

Sous l'hypothèse monochromatique, nous pouvons réécrire l'équation (1.1) comme :

$$N \sim \mathcal{P}\left(n_0 \, \mathrm{e}^{-\int_{L_i} \mu(x) \, \mathrm{d}\, x}\right), \tag{1.2}$$

où $n_0$ est simplement l'intensité d'émission maximale de la source.

### 1.1.3 Les méthodes analytiques en bref

Afin de faciliter la compréhension, les méthodes analytiques sont présentées dans un cadre bidimensionnel et peuvent être généralisées en trois dimensions. Bien que nous ayons présenté l'équation de Beer-Lambert dans un cadre *probabiliste*, les méthodes analytiques, elles, n'utilisent pas cette information. Nous pouvons simplement supposer que l'intensité mesurée au détecteur est égale à la valeur moyenne prédite par (1.2) et obtenir la loi de Beer-Lambert *déterministe* monochromatique :

$$I(L_i) = I_0 \, \mathrm{e}^{-\int_{L_i} \mu(x) \, \mathrm{d}\, x}, \tag{1.3}$$

où $I(L_i)$ désigne l'intensité mesurée à un détecteur pour un rayon suivant un parcours $L_i$ et $I_0$ est l'intensité incidente de la source. Nous introduisons cette nouvelle notation afin d'éviter toute confusion entre le cadre déterministe et le cadre stochastique.

Dans un cadre mathématique bidimensionnel où on considère la loi monochromatique (1.3), la *tranche* de l'objet est représentée par $\mu(x_1, x_2) : \mathbb{R}^2 \to \mathbb{R}$. Pour l'instant, on se limite au cas où la source émet des rayons parallèles. En prenant comme référentiel la barrette de détecteurs, on peut représenter toute mesure d'intensité du faisceau transmis par un couple $(\rho, \theta)$, avec $\rho \in \mathbb{R}^{n_d}$, où $n_d$ est le nombre de détecteurs, et $\theta \in \{[0, \pi[\}^{n_\theta}$, où $n_\theta$ est la discrétisation angulaire. L'équation d'une droite reliant la source à un détecteur pour un angle $\theta$ donnée peut donc être formulée selon $L(\rho, \theta)$ et on peut ainsi réécrire la loi de Beer-

Lambert précédente (1.3) comme :

$$I(\rho,\theta) = I_0\,\mathrm{e}^{-\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mu(x_1,x_2)\delta(x_1\cos\theta+x_2\sin\theta-\rho)\,\mathrm{d}\,x_1\mathrm{d}\,x_2}\,. \tag{1.4}$$

Les quantités définies précédemment sont illustrées à la figure 1.2. Nous nous intéressons maintenant à reconstruire les valeurs de $\mu(x_1, x_2)$ depuis des mesures d'intensité données.

### Rétroprojection filtrée (*Filtered Backprojection*)

En prenant le logarithme de (1.4) et en introduisant la fonction $y : \mathbb{R}^{1\times[0,\ \pi[} \to \mathbb{R}$ telle que $y(\rho,\theta) = \ln\left(\frac{I_0}{I(\rho,\theta)}\right)$, on définit une *projection* pour $(\rho, \theta)$ :

$$y(\rho,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mu(x_1,x_2)\delta(x_1\cos\theta + x_2\sin\theta - \rho)\,\mathrm{d}\,x_1\mathrm{d}\,x_2, \tag{1.5}$$

où $\delta$ est la distribution de Dirac. Intuitivement, on peut remarquer qu'une projection est limitée à la partie de l'objet $\mu(x_1, x_2)$ traversée par le rayon propre à la droite $L(\rho, \theta)$, ce qui corrobore les explications présentées à la section 1.1.1. La collection $y(\rho,\theta)\ \forall\ \rho,\ \theta$, c'est-à-dire l'ensemble des projections, est communément appelée le *sinogramme*. Le côté droit de l'équation (1.5) correspond en fait à la *transformée de Radon* de $\mu(x_1, x_2)$, voir, e.g., (Prince et Links, 2007), de sorte que l'on peut simplement réécrire $y(\rho,\theta) = \mathcal{R}\{\mu(x_1, x_2)\}$. On peut d'ores et déjà conclure qu'il est possible d'obtenir $\mu(x_1, x_2)$ directement, pourvu qu'une inverse à l'opérateur $\mathcal{R}$ existe.

Cela est possible par l'entremise du théorème de la *tranche de Fourier*. Ce théorème stipule que la transformée de Fourier de toutes les projections à un angle $\theta$ est une *ligne* à un angle $\theta$ de la transformée de Fourier de l'objet que l'on désire imager. Ce résultat est illustré à la figure 1.3 et nous invitons le lecteur à se rapporter à, e.g., (Prince et Links, 2007) pour plus de détails.

En définissant $Y(\omega,\theta)$ la transformée de Fourier du sinogramme sur $\rho$ et en substituant la définition de $y(\rho,\theta)$ (1.5) on obtient :

$$\begin{aligned} Y(\omega,\theta) &= \int_{-\infty}^{\infty} y(\rho,\theta)\,\mathrm{e}^{-2\pi\mathrm{i}\,\omega\rho}\,\mathrm{d}\,\rho, \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\mu(x_1,x_2)\,\mathrm{e}^{-2\pi\mathrm{i}\,\omega(x_1\cos\theta+x_2\sin\theta)}\,\mathrm{d}\,x_1\mathrm{d}\,x_2, \\ &= \mathscr{F}\left[\mu(x_1,x_2)\right], \end{aligned}$$

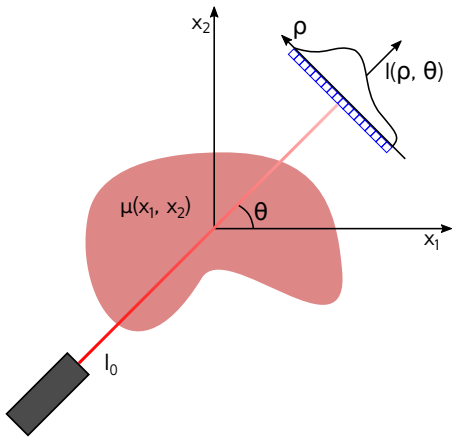



Figure 1.2 Schéma de la tomographie pour une source avec rayons paralèlles

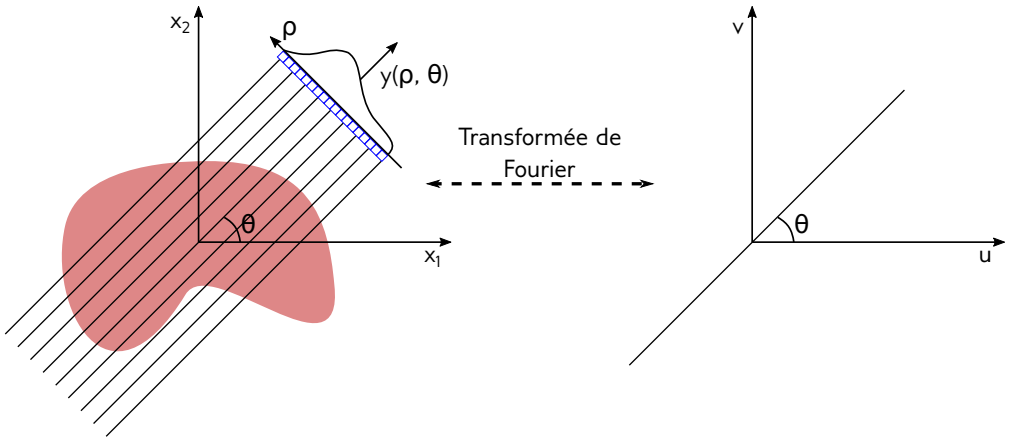



Figure 1.3 Schéma du théorème de la tranche de Fourier appliqué à $y(\rho, \theta)$

où on peut isoler $\mu(x_1, x_2)$ pour obtenir :

$$\begin{aligned}\mu(x_1, x_2) &= \int_0^{2\pi}\int_0^{\infty} \mathscr{F}(\omega\cos\theta, \omega\sin\theta)\, \mathrm{e}^{2\pi \mathrm{i}\,\omega(x_1\cos\theta + x_2\sin\theta)}\, \omega\,\mathrm{d}\,\omega\,\mathrm{d}\,\theta, \\ &= \int_0^{2\pi}\int_0^{\infty} Y(\omega,\theta)\, \mathrm{e}^{2\pi \mathrm{i}\,\omega(x_1\cos\theta + x_2\sin\theta)}\, \omega\,\mathrm{d}\,\omega\,\mathrm{d}\,\theta, \\ &= \int_0^{\pi}\Big[\int_{-\infty}^{\infty} |\omega| Y(\omega,\theta)\, \mathrm{e}^{2\pi \mathrm{i}\,\omega\rho}\,\mathrm{d}\,\omega\Big]_{\rho = x_1\cos\theta + x_2\sin\theta} \mathrm{d}\,\theta.\end{aligned}$$

Ce résultat à lui seul justifie l'intérêt des manufacturiers de tomographes envers les méthodes analytiques : il est possible de reconstruire une image par le calcul d'une transformée de Fourier du sinogramme et une intégrale sur tous les angles. Ce sont toutes deux des opérations peu coûteuses d'un point de vue numérique. De plus, lors de la mise en œuvre, les transformées de Fourier sont accomplies par FFTs (*Fast Fourier Transforms*), faisant en sorte que le nombre d'opérations requis est moindre. Le lecteur peut se référer à Cooley et Tukey (1965) au sujet des opérations de FFTs. On note également l'apparition naturelle de $|\omega|$ dans l'expression finale qui agit comme un filtre rampe, d'où le nom *rétroprojection filtrée*.

### 1.1.4 Les méthodes itératives statistiques en bref

Dans le cadre des méthodes itératives statistiques, l'information probabiliste, telle que celle contenue dans (1.2), est interprétée afin de mener ultimement à la reconstruction d'une image. Nous détaillons également comment une *discrétisation* de l'intégrand dans (1.2) permet d'obtenir une modélisation adéquate aux calculs numériques. Le lecteur peut se référer aux travaux de Sauer et Bouman (1993), Bouman et Sauer (1993) et Fessler (2000) pour de plus amples détails quant aux modélisations probabilistes du problème de reconstruction tomographique.

#### Loi de Beer-Lambert stochastique discrétisée

En dépit de l'approximation monochromatique, l'expression (1.2) demeure peu appropriée aux calculs numériques, principalement à cause de la dépendance spatiale de $\mu$. Sauer et Bouman (1993) remédient à ce problème en *discrétisant* le domaine de $\mu$, grâce à l'introduction d'une *fonction de discrétisation* $\xi(x)$. Cela permet d'assigner un coefficient d'atténuation constant $\mu_j, j = 1, \ldots, n_{\text{vox}}$ à un $j^{\text{ième}}$ voxel, où $n_{\text{vox}}$ est le nombre de voxels discrétisés. La fonction de discrétisation peut être interprétée comme un maillage arbitraire tridimensionnel, défini en fonction de notre choix de coordonnées, tel qu'illustré schématiquement à la figure 1.4. Au cours de notre étude, nous évaluons spécifiquement les bénéfices de l'utilisation de coordonnées cylindriques par rapport aux coordonnées cartésiennes. Le lecteur peut se ré-

férer à l'annexe A pour une description du calcul de la matrice de projection en coordonnées cylindriques.

Le résultat de l'intégrale de (1.2) correspond alors à une *collection des longueurs d'intersection* entre un rayon suivant la trajectoire $L_i$ et les voxels qu'il croise. Mathématiquement, on peut résumer ces idées de la manière suivante, où $j$ correspond à un voxel et $i$ à une mesure :

$$\begin{aligned}\int_{L_i} \mu(x)\,\mathrm{d}\,x &= \int_{L_i} \sum_j \mu_j \xi_j(x)\,\mathrm{d}\,x, \\ &= \sum_j \mu_j \underbrace{\int_{L_i} \xi_j(x)\,\mathrm{d}\,x}_{p_{ij}}, \\ &= \sum_j p_{ij}\mu_j.\end{aligned}$$

Nous définissons $\mu \in \mathbb{R}^{n_{\text{vox}}}$, le vecteur des coefficients d'atténuation, et $P \in \mathbb{R}^{n_{\text{meas}} \times n_{\text{vox}}}$, la *matrice de projection*. D'emblée, on peut observer que la résolution de l'image, et donc la taille de l'inconnu $\mu$ et de la matrice de projection, sont tous deux directement corrélés à notre discrétisation de l'espace. En utilisant cette nouvelle notation, on obtient une formulation matrice-vecteur de (1.2) qui est maintenant propice aux calculs numériques :

$$N \sim \mathcal{P}\left(n_0\,\mathrm{e}^{-P\mu}\right). \tag{1.6}$$

En supposant que les comptes de photons sont indépendants et identiquement distribués, Sauer et Bouman (1993) définissent la distribution de probabilité conditionnelle de (1.6) selon :

$$P(N = n \,|\, \mu) = \prod_i \left[\frac{\exp\left(-n_0\,\mathrm{e}^{-[P\mu]_i}\right)\left(n_0\,\mathrm{e}^{-[P\mu]_i}\right)^{n_i}}{n_i!}\right], \tag{1.7}$$

Nous avons dorénavant une expression permettant d'évaluer la probabilité de mesurer les comptes de photon $n$ pour une distribution de coefficients d'atténuation $\mu$ donnée. Il est donc intuitif de supposer que le *vrai* $\mu$ est la valeur la plus *probable* de la distribution (1.7). Cette procédure d'estimation correspond en fait à une approche de *maximum de vraisemblance* (*maximum likelihood*).

### 1.1.5 Maximum de vraisemblance et maximum a posteriori

Le principe du maximum de vraisemblance (ML) est d'identifier la valeur qui maximise une fonction de densité de probabilité conditionnelle. En considérant (1.7), on obtient ainsi le

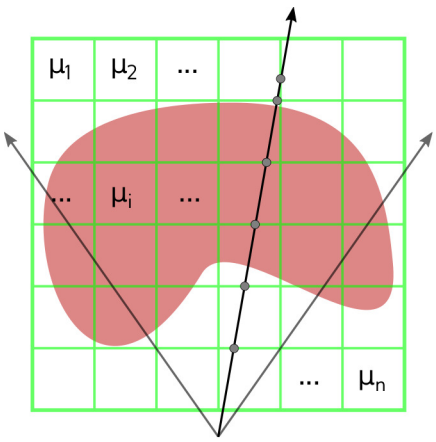



Figure 1.4 Schéma d'une discrétisation cartésienne du domaine de $\mu$, illustré en 2D

meilleur estimateur au sens du maximum de vraisemblance :

$$\hat{\mu}_{\mathrm{ML}} = \underset{\mu \geq 0}{\operatorname{argmax}} \; P(N = n \,|\, \mu), \tag{1.8}$$

où nous incluons une contrainte de positivité sur $\mu$, car $\mu$ ne peut physiquement pas être négatif. En raison de la nature de $\mu$, il est possible que nous ayons une idée de la forme générale des solutions, et ce, avant même de résoudre (1.8). Ce type d'information est appelé *densité de probabilité a priori* $P(\mu)$ et provient d'une modélisation adaptée au cadre dans lequel le problème (1.8) est posé. Dans le cadre d'un processus d'estimation, nous pouvons exploiter cette information supplémentaire par l'entremise de la règle de Bayes :

$$P(\mu \,|\, n) = \frac{P(n \,|\, \mu) P(\mu)}{P(n)}. \tag{1.9}$$

On obtient donc le nouvel estimateur au sens du maximum a posteriori :

$$\hat{\mu}_{\mathrm{MAP}} = \underset{\mu \geq 0}{\operatorname{argmax}} \; \frac{P(n \,|\, \mu) P(\mu)}{P(n)} \tag{1.10}$$

où on pondère notre probabilité conditionnelle par notre supposition a priori et où on impose également la contrainte de positivité sur $\mu$. L'estimateur (1.10) réalise donc un compromis entre une solution qui permet de prédire les données observées et une solution qui est fidèle à notre modèle a priori. De manière générale, si on s'intéresse uniquement à la valeur de l'estimateur $\hat{\mu}$, on peut exclure le dénominateur de la fonction objectif dans (1.10), car il ne dépend pas de $\mu$.

Depuis les problèmes généraux (1.8) et (1.10), plusieurs choix de modélisation supplémen-

taires peuvent être effectués afin de faciliter la résolution du problème. À la section suivante, nous présentons l'approche adoptée par Hamelin (2009), les raffinements proposés par Goussard *et al.* (2013) ainsi que Golkar (2013) et démontrons comment cela mène au problème d'optimisation que nous chercherons à résoudre. Pour plus d'informations sur le maximum de vraisemblance et le maximum a posteriori, le lecteur peut se référer à, e.g., (Navidi, 2010).

## 1.2 Éléments de la problématique

En reprenant l'expression (1.7), on dénote d'emblée la complexité de l'un ou l'autre des problèmes d'optimisation obtenus, soit (1.8) ou (1.10), étant donnée la nature fortement non-linéaire de la loi de Poisson. Puisque la fonction (1.7) est de forme exponentielle, une approche commune est de maximiser le négatif de son logarithme (communément appelé *negative log-likelihood*), ce qui permet d'éliminer une partie de la non-linéarité sans altérer les résultats :

$$L(n \,|\, \mu) = \sum_i \left[ n_0 \, \mathrm{e}^{-[P\mu]_i} + n_i [P\mu]_i + \log(n_i!) \right]. \tag{1.11}$$

En définissant le *sinogramme* $y$ de la même manière qu'à la section 1.1.3 :

$$y = \ln\left(\frac{n_0}{n}\right), \tag{1.12}$$

où les opérations sont effectuées terme à terme, on peut réécrire (1.6) sous la forme d'un système linéaire :

$$y = P\mu. \tag{1.13}$$

Sauer et Bouman (1993) effectuent un développement de Taylor d'ordre deux pour chaque terme de (1.11) autour de $y$, de sorte à obtenir une fonction objectif aux moindres carrés :

$$\begin{aligned} L(n \,|\, \mu) &= \sum_i \left[ n_0 \, \mathrm{e}^{-[P\mu]_i} + n_i [P\mu]_i + \log(n_i!) \right], \\ &\approx \sum_i \left[ n_i(1 + y_i) + \log(n_i!) + \tfrac{n_i}{2} \left( [P\mu]_i - y_i \right)^2 \right], \\ &= \tfrac{1}{2} \left\| P\mu - y \right\|_{\Delta_N}^2 + \sum_i \left[ n_i(1 + y_i) + \log(n_i!) \right], \end{aligned} \tag{1.14}$$

où $\Delta_N$ est défini de sorte que $\Delta_N = \mathrm{diag}\,(n_i) \;\; \forall \; i = 1, \; \ldots, \; n_{\mathrm{meas}}$.

Puisqu'un $n_i$ plus élevé signifie que la projection passe par une région de l'objet plus *transparente* aux rayons X, la matrice $\Delta_N$ permet d'améliorer le rapport signal sur bruit en compensant les projections qui ont subi moins d'atténuation. En substituant (1.14) dans

l'équation (1.8) et en éliminant les termes qui ne dépendant pas de $\mu$, le meilleur estimateur de maximum vraisemblance devient :

$$\hat{\mu}_{\mathrm{ML}} = \underset{\mu \geq 0}{\operatorname{argmin}} \ \tfrac{1}{2} \left\| P\mu - y \right\|_{\Delta_N}^2. \tag{1.15}$$

Le choix d'utiliser l'approximation (1.14) est évidemment arbitraire, mais permet d'obtenir une fonction objectif *convexe*, voir, e.g., (Nocedal et Wright, 2000).

Par observation de l'estimateur (1.15), on peut conclure que le développement (1.14) revient à approximer la loi de Poisson par une loi normale, tel que le *théorème central limite* stipule lorsque le nombre de mesures est élevé. Une dérivation complète est présentée à l'annexe D. Alternativement, nous pouvons poser $\Delta_N = I$ dans (1.15) afin de simplifier d'avantage la fonction objectif, de sorte qu'on obtient le problème :

$$\hat{\mu}_{\mathrm{ML}} = \underset{\mu \geq 0}{\operatorname{argmin}} \ \tfrac{1}{2} \left\| P\mu - y \right\|^2. \tag{1.16}$$

Pour la suite de cet ouvrage, nous considérons cette simplification, mais nous gardons toujours la possibilité de raffiner notre modèle en utilisant $\Delta_N = \operatorname{diag}(n)$. D'autre part, si on désire utiliser un estimateur au sens du maximum a posteriori, on peut conclure que l'utilisation de tout $P(\mu)$ de la forme :

$$P(\mu) \propto \mathrm{e}^{-\lambda \phi(\mu)},$$

dans (1.10), correspond en fait à ajouter un terme de *pénalité* à la fonction objectif de (1.16). Rappelons que nous considérons le cadre de la neg-log-vraisemblance présenté précédemment. En conséquence, on obtient l'estimateur MAP :

$$\begin{aligned} \min_{\mu} \quad & \tfrac{1}{2} \left\| P\mu - y \right\|^2 + \lambda \phi(\mu) \\ \text{s.c.} \quad & \mu \geq 0. \end{aligned} \tag{OP}$$

D'un point de vue numérique, il est préférable d'introduire une fonction de pénalité qui permet de préserver la *convexité* de la fonction objectif, car cela assure que tout point stationnaire est un minimum global. D'ailleurs, si la convexité n'est pas préservée, l'approximation (1.14) aura été effectuée en vain. Puisque nous reconstruisons une image constituée de coefficients d'atténuation propres à des tissus biologiques, nous cherchons à pénaliser les très fortes variations locales de $\mu$. En effet, dans le corps humain, les régions de tissu sont plutôt homogènes. À cette fin, Goussard *et al.* (2013) utilisent des fonctions de pénalisation telles que la norme

$\mathscr{L}_2$ :

$$\phi_{\mathscr{L}_2}(\mu) = \tfrac{1}{2} \sum_{n=1}^{n_{\text{dim}}} \mu^{\mathsf{T}} {D^{(n)}}^{\mathsf{T}} \Gamma^{(n)} D^{(n)} \mu, \tag{1.17}$$

et la norme $\mathscr{L}_2\,\mathscr{L}_1$ :

$$\phi_{\mathscr{L}_2\,\mathscr{L}_1}(\mu) = \sum_{n=1}^{n_{\text{dim}}} e^{\mathsf{T}} \Gamma^{(n)} \left( \delta^2 e + (D^{(n)}\mu)^2 \right)^{1/2}, \tag{1.18}$$

où $e \in \mathbb{R}^{n_{\text{vox}}}$ est le vecteur constitué de 1 et $\delta \in \mathbb{R}$ est un paramètre. Les matrices jacobiennes $\Gamma^{(n)} \in \mathbb{R}^{n_{\text{vox}} \times n_{\text{vox}}}$, $n = 1, \ldots, n_{\text{dim}}$ sont des matrices diagonales introduites afin de pondérer la pénalisation associée à un voxel en fonction de sa taille. On introduit également les *matrices de différences finies*, notée $D^{(n)} \in \mathbb{R}^{n_{\text{vox}} \times n_{\text{vox}}}$ avec $n = 1, \ \ldots, \ n_{\text{dim}}$, qui permettent de pénaliser selon la valeur des voxels voisins d'un voxel donné pour chaque direction spatiale. On peut retrouver les fonctions de pénalité $\mathscr{L}_2$ et $\mathscr{L}_2\,\mathscr{L}_1$ usuelles en posant $D^{(n)} = I_{n_{\text{vox}}}$ $\forall$ $n$, i.e. seule la valeur d'un voxel donné influence sa pénalité. Dans le cas d'une pénalisation $\mathscr{L}_2$, nous référons au premier cas en tant que *pénalisation sur le gradient de l'objet* et au second en tant que *pénalisation sur l'objet*.

En procédant à des expériences numériques sur (OP), Goussard *et al.* (2013) et Golkar (2013) ont conclu que l'utilisation de coordonnées cartésiennes menait à des matrices de projection très lourdes à stocker en mémoire. Afin de remédier à ce problème, Goussard *et al.* (2013) ont étudié la représentation en coordonnées cylindriques et ont démontré que cela permettait de bénéficier de la symétrie du processus d'acquisition des données. Cette symétrie se manifeste dans notre problème par un opérateur de projection $P$ bloc-circulant, de sorte que les propriétés présentées à l'annexe B peuvent être exploitées afin de réduire drastiquement le stockage mémoire requis. Le lecteur peut se rapporter à l'annexe A pour de plus amples détails sur le calcul de la matrice de projection en coordonnées cylindriques. Néanmoins, Goussard *et al.* (2013) ont aussi constaté que les solveurs appliqués au problème (OP) en coordonnées cylindriques convergent difficilement pour les voxels centraux, de sorte que les images reconstruites ne sont pas adéquates. Cette difficulté est causée par les fortes différences de taille entre les voxels, qui se manifestent par un piètre conditionnement de $P$. En pratique, de tels problèmes sont surmontés par l'utilisation d'un *préconditionneur*.

### 1.2.1 Le problème mis à l'échelle en coordonnées cylindriques

Pour pallier les difficultés de convergence près de l'origine rencontrées en coordonnées cylindriques, Golkar (2013) a étudié plusieurs *matrices de mise à l'échelle* pour le problème (OP). Au cours de cet ouvrage, nous nous contentons d'utiliser la matrice jugée la plus performante

et référons le lecteur à (Golkar, 2013) pour plus de détails. La *matrice de mise à l'échelle*, notée $C$, que nous considérons dans cet ouvrage est de la forme :

$$C = \tfrac{1}{n}\, \mathscr{F}_n^\star\, \Delta\, \mathscr{F}_n, \tag{1.19}$$

où $\mathscr{F}_n$ représente une matrice de Fourier, $n = n_{\text{vox}}$ et $\Delta$ est une matrice diagonale. Le lecteur peut se rapporter à l'annexe B pour plus d'informations sur les matrices de Fourier et à l'annexe C pour une dérivation complète de (1.19). Lors de la mise en oeuvre, les produits avec les matrices de Fourier peuvent être effectués par des FFTs (Cooley et Tukey, 1965) qui ne requièrent que $\mathcal{O}(n \log n)$ opérations. Idéalement, nos solveurs devraient tirer avantage du faible coût d'un produit matrice-vecteur avec $C$.

La matrice (1.19) peut être interprétée comme un *préconditionneur*, car elle diminue l'étalement spectral du hessien du problème (OP). Par contre, nous y référons en tant que *matrice de mise à l'échelle*, puisqu'elle est introduite dans (OP) par le changement de variable :

$$\mu = Cx. \tag{1.20}$$

On obtient ainsi le problème d'optimisation *mis à l'échelle* :

$$\begin{aligned} \min_{x} \quad & \tfrac{1}{2}\left\| PCx - y \right\|^2 + \lambda\phi(Cx) \\ \text{s.c.} \quad & Cx \geq 0. \end{aligned} \tag{SP}$$

Un inconvénient de la mise à l'échelle (1.20) est mis en évidence dans le problème (SP) : la contrainte de positivité est transformée en contrainte d'inégalités linéaires, augmentant considérablement la complexité du problème. D'autre part, le solveur L-BFGS-B, recommandé par Hamelin (2009) pour le problème (OP), n'est donc plus applicable au problème (SP). Cela soulève donc la question qui motive nos travaux, à savoir quelles méthodes numériques sont applicables et performantes pour le problème (SP).

## 1.3 Objectifs de recherche

Cet ouvrage est consacré à l'étude et à l'évaluation de la performance de différents algorithmes sur le problème de reconstruction d'image (SP). Nous cherchons à développer des méthodes qui puissent bénéficier des gains en stockage mémoire entraînés par la discrétisation en coordonnées cylindrques, tout en ayant des temps d'exécution propices à l'application clinique.

D'emblée, nous faisons la distinction entre deux familles de méthodes d'optimisation : les

méthodes de premier et de deuxième ordre. Comme le laisse présager leurs noms, les méthodes de premier ordre n'ont recours qu'à l'information du gradient, tandis que les méthodes de deuxième ordre utilisent également le hessien. Dans le cadre de notre problème, nous considérons toutes deux, en raison du compromis entre vitesse de convergence et coût computationnel. La taille de notre problème étant considérable, nous cherchons particulièrement à identifier des méthodes d'optimisation *sans factorisation* pour problèmes convexes avec contraintes.

Dans le domaine de l'optimisation, on traite généralement des problèmes sous contraintes avec des méthodes de points intérieurs, de Lagrangien augmenté ou de contraintes actives. En vertu des propriétés particulières du jacobien de nos contraintes d'inégalités linéaires, nous émettons l'hypothèse qu'il est possible de développer des projections efficaces sur l'ensemble réalisable. Cette hypothèse s'avérant vraie, il est alors fort probable que les méthodes de contraintes actives à base de projections soient les plus performantes sur le problème (SP). Pour cette raison, nous concentrons notre attention sur ces dernières et cherchons à les implémenter efficacement. Néanmoins, les méthodes de points intérieurs sont généralement appropriées pour ce type de problème, de sorte que cette alternative demeure sujette à l'étude.

## 1.4 Plan du mémoire

La suite de cet ouvrage est divisée comme suit. D'abord, nous introduisons les méthodes de Krylov, qui permettent de résoudre des systèmes linéaires sans factorisation, pour ensuite présenter les fondements théoriques de l'optimisation avec contraintes. Nous synthétisons l'ensemble de ces concepts en faisant une description générique de la famille d'algorithmes que nous avons considérée dans notre étude, soit les méthodes de contraintes actives. Plus particulièrement, nous faisons une description exhaustive d'une méthode de contraintes actives à base de projections, soit le solveur TRON, que nous avons cherché à adapter au problème (SP). Cette variante de TRON, en plus d'une méthode de gradient projeté spectral, sont détaillées et appliquées au problème (SP) dans notre article à la section 4. Nous terminons par une critique de nos travaux ainsi qu'une discussion des améliorations possibles et des pistes additionnelles à considérer. Soulignons qu'une description générale des méthodes de points intérieurs est présentée à l'annexe E en raison de leur pertinence dans nos travaux.

# CHAPITRE 2 REVUE DE LITTÉRATURE

Cette section se veut un résumé des méthodes de Krylov ainsi que du formalisme mathématique associé à l'optimisation avec contraintes. Nous proposons également une description générale des méthodes de contraintes actives et une présentation exhaustive de l'algorithme TRON.

## 2.1 Systèmes linéaires et méthodes de Krylov

Au cours de notre étude, nous sommes amenés à manipuler et résoudre des systèmes matriciels de taille importante. Une conséquence de la taille imposante et de la densité de nos systèmes est l'impossibilité de les stocker en mémoire et donc de calculer une *factorisation*. Les méthodes de Krylov sont des méthodes permettant de résoudre un système linéaire $Ax = b$ en possédant uniquement l'opération de produit matrice-vecteur avec $A$ et éventuellement $A^\mathsf{T}$.

### 2.1.1 Méthode du gradient conjugué

La méthode du gradient conjugué (*conjugate gradient method*, abrégé CG) est une méthode conçue spécialement pour le cas où $A$ est symétrique et définie positive, soit $A = A^\mathsf{T}$ et $v^\mathsf{T}Av > 0$ pour tout vecteur $v$ non nul. La particularité de CG est la génération de directions conjuguées $p_i$ par rapport à la matrice $A$, c'est-à-dire que $p_i^\mathsf{T}Ap_j = 0$ pour $j \neq i$. L'intuition derrière CG provient du fait que le problème $Ax = b$ représente les conditions d'optimalité du problème d'optimisation quadratique sans contraintes (Hestenes et Stiefel, 1952)

$$\min_x \ \phi(x) := \tfrac{1}{2}x^\mathsf{T}Ax - b^\mathsf{T}x,$$

de sorte qu'on peut obtenir le minimum de $\phi(x)$ en résolvant

$$\nabla_x\phi(x) = Ax - b = 0 =: r(x).$$

En particulier, puisque des directions conjuguées sont linéairement indépendantes, on peut minimiser $\phi(x)$ en minimisant successivement selon les $p_j$. On génère itérativement la suite

$$x_{k+1} = x_k + \alpha_k p_k,$$

en faisant une recherche linéaire exacte, c'est-à-dire en prenant $\alpha_k$ tel que $\frac{d\phi(x_k+\alpha_k\rho_k)}{d\alpha_k} = 0$. On obtient ainsi

$$\alpha_k = -\frac{r_k^\mathsf{T} p_k}{p_k^\mathsf{T} A p_k},$$

où

$$r_k = Ax_k - b$$

est le résidu. Finalement, Hestenes et Stiefel (1952) génèrent la direction $p_{k+1}$ par une combinaison linéaire de $\nabla\phi(x_k) = r_k$ et $p_k$. Au théorème 4.1, (Hestenes et Stiefel, 1952) démontrent que cela garantit $p_{k+1}^\mathsf{T} A p_k = 0 \ \forall \ k$ et même $p_i^\mathsf{T} A p_j = 0 (i \neq j)$.

L'algorithme 1 détaille la méthode du gradient conjugué telle que développée par Hestenes et Stiefel (1952). Typiquement, une condition d'arrêt basée sur la norme du résidu $\left\|r_k\right\|$ est

**Algorithme 1** Méthode du gradient conjugué

1: Initialiser $x_0 = 0$, $r_0 = b$ et $p_0 = r_0$
2: **pour** k = 1, 2, 3, ... **faire**
3: $\quad \alpha_k = \frac{r_{k-1}^\mathsf{T} r_k}{p_{k-1}^\mathsf{T} A p_{k-1}}$
4: $\quad x_k = x_{k-1} + \alpha_k p_{k-1}$
5: $\quad r_k = r_{k-1} - \alpha_k A p_{k-1}$
6: $\quad \beta_k = \frac{r_k^\mathsf{T} r_k}{r_{k-1}^\mathsf{T} r_{k-1}}$
7: $\quad p_k = r_k + \beta_k p_{k-1}$
8: **fin pour**

utilisée.

### 2.1.2 Méthode du résidu minimal

L'algorithme de MINRES, pour *Minimum Residual*, est basé sur l'itération de Lanczos, qui a pour but de transformer une matrice sous forme tridiagonale. Cette transformation permet de calculer récursivement une base orthonormale de l'image de $A$, qui pourra par la suite être utilisée pour résoudre le système $Ax = b$. L'intuition derrière MINRES est donc similaire à celle du gradient conjugué, mais dans un cas plus général. Cette méthode requiert que $A$ soit hermitienne, c'est-à-dire que $A^\star = A$, où $(\cdot)^\star$ dénote l'opération de transposée conjuguée. Il est important de noter que CG est aussi basé sur l'itération de Lanczos, bien que Hestenes et Stiefel (1952) l'aient développé autrement.

Paige et Saunders (1975) utilisent la relation entre $A$ et la matrice tridiagonale symétrique $T$ établie par Lanczos :

$$A = QTQ^\star, \tag{2.1}$$

où $Q$ est une matrice unitaire, i.e. $Q^\star Q = I$. Puisque $T$ est symétrique et tridiagonale, on peut la représenter selon les vecteurs $\alpha$ et $\beta$ constituant sa diagonale principale et une de ses diagonales secondaires. On définit les sous matrices $A_k$, $Q_k$ et $T_k$ respectivement constituées des $k$ premières lignes et colonnes de $A$, $Q$ et $T$. Pour $k$ donné, nous pouvons écrire les matrices

$$T_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{bmatrix},$$

et

$$\tilde{T}_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \\ & & & & \beta_{k+1} \end{bmatrix}, \tag{2.2}$$

où $\tilde{T}_k$ possède une ligne de plus que $T_k$. À l'aide de ces deux matrices, Paige et Saunders (1975) constatent que la relation de récurrence :

$$Aq_k = \beta_k q_{k-1} + \alpha_k q_k + \beta_{k+1} q_{k+1} \tag{2.3}$$

peut être utilisée pour obtenir $q_{k+1}$. Cette procédure permettant de générer une base orthonormale porte le nom d'itération de Lanczos et est formalismée selon l'algorithme 2 (Trefethen et Bau III, 1997). On peut remarquer que cela revient en fait à appliquer l'algorithme de Gram-Schmidt à une matrice symétrique.

**Algorithme 2** Itération de Lanczos

1: Initialiser $\beta_0 = 0$, $q_0 = 0$, $\beta_1 = \left\|b\right\|$ et $\beta_1 q_1 = b$ avec $b$ donné.
2: **pour** k = 1, 2, 3, ... **faire**
3: $\quad v = Aq_k$
4: $\quad \alpha_k = q_k^{\mathsf{T}} v$
5: $\quad v = v - \beta_{k-1} q_{k-1} - \alpha_k q_k$
6: $\quad \beta_{k+1} q_{k+1} = v - \alpha_k q_k - \beta_k q_{k-1}$
7: **fin pour**

Une propriété importante de l'itération de Lanczos est que les $q_k$ qui sont générés sont dans le sous-espace de Krylov formé par $A$ et $b$, voir, e.g., (Trefethen et Bau III, 1997) :

$$\mathcal{K}_k(A, b) = \langle b, Ab, \ldots, A^{k-1}b \rangle,$$

où la notation $\langle a, b, \ldots \rangle$ désigne l'espace formé par les vecteurs $a, b, \ldots$. Nous pouvons donc substituer tout $x_k$, une approximation de la solution de $Ax = b$, par une combinaison linéaire des vecteurs $q_i$, car $x_k$ est aussi un vecteur de $\mathcal{K}_k(A, b)$ :

$$x_k = Q_k y. \tag{2.4}$$

Plus précisément, nous cherchons une solution approchée sous la forme d'une combinaison des $q_k$. Considérons maintenant le problème aux moindres carrés :

$$\min_{x_k} \ \left\| Ax_k - b \right\|^2.$$

En substituant (2.4) et (2.1) dans le problème aux moindres carrés, on obtient

$$\min_{y} \ \left\| Q_{k+1} \tilde{T}_k y - b \right\|^2.$$

Finalement, en multipliant à l'intérieur de la norme par $Q_{k+1}^\star$, on peut obtenir une formulation plus intéressante :

$$\min_{y} \ \left\| \tilde{T}_k y - \beta_1 e_1 \right\|^2, \tag{2.5}$$

en notant que $q_1$ est initialisé à $b/||b||$, que tous les $q_i$ sont orthogonaux et où $e_1$ est le vecteur colonne constitué d'un 1 suivi de 0.

MINRES peut alors être formalisé suivant l'algorithme 3, tel que proposé par Paige et Saunders (1975). En pratique, MINRES n'utilise pas réellement (2.4), mais plutôt une mise à jour

**Algorithme 3** MINRES

1: Initialiser $q_1 = b/||b||$ avec $b$ donné
2: **pour** k = 1, 2, 3, ... **faire**
3: Étape $k$ de l'itération de Lanczos (algorithme 2)
4: Résolution du problème (2.5)
5: Calcul de $x_k$ à partir de $y$ selon (2.4)
6: **fin pour**

de la forme $x_{k+1} = x_k + \phi_k w_k$ qui évite de mémoriser tous les $q_k$. Pour plus de détails, nous référons le lecteur à (Paige et Saunders, 1975).

### 2.1.3 Remarques sur LSQR et LSMR

Les algorithmes LSQR (Paige et Saunders, 1982) et LSMR (Fong et Saunders, 2011) sont respectivement des variantes de MINRES et CG. Leur particularité est de traiter les équations normales

$$A^T A x = A^T b \tag{2.6}$$

résultant des conditions de $\mathcal{KKT}$ d'ordre un du problème aux moindres carrés

$$\min_x \ \|Ax - b\|^2$$

plutôt que $Ax = b$, car si $A$ est rectangulaire, ce système n'a peut-être aucune solution. L'équation (2.6) est communément appelée l'*équation normale* du problème aux moindres carrés. LSMR et LSQR permettent également d'utiliser un terme de régularisation si la matrice $A$ est mal conditionnée, menant au problème suivant :

$$\min_x \ \left\| \begin{bmatrix} A \\ \lambda I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|^2,$$

où $\lambda$ est un paramètre de régularisation. Elles sont par construction équivalentes à CG et MINRES — en arithmétique exacte — en plus d'être plus stables numériquement. Puisque nous avons déjà fait un survol de CG et MINRES, nous référons aux travaux de Paige et Saunders (1982) et Fong et Saunders (2011) pour de plus amples détails.

## 2.2 Optimisation sous contraintes

Au cours de cette section, un problème d'optimisation sous contraintes général, ainsi que les conditions d'optimalité de premier ordre qui lui sont associées, sont détaillés. En optimisation *continue*, on cherche à obtenir le minimum ou le maximum d'une fonction objectif différentiable non-linéaire, notée $f(x)$, sous un certain ensemble de contraintes. Cet ensemble peut être une mixture d'égalités et d'inégalités différentiables. En conséquence, on écrit un problème général sous la forme :

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.c.} \quad & c_i(x) \geq 0, \quad i \in \mathcal{I}, \\ & c_i(x) = 0, \quad i \in \mathcal{E}, \end{aligned} \tag{2.7}$$

où $\mathcal{E}$ et $\mathcal{I}$ indexent respectivement les contraintes d'égalité et d'inégalité. Par observation du problème (2.7), un point *admissible*, ou *réalisable*, est un point de l'ensemble

$$\Omega = \Big\{ x \in \mathbb{R}^n \;\Big|\; c_i(x) \geq 0,\ i \in \mathcal{I},\ c_i(x) = 0,\ i \in \mathcal{E} \Big\}. \tag{2.8}$$

En conséquence, on peut définir l'index des contraintes actives en un point arbitraire $x \in \Omega$ comme étant

$$\mathcal{A}(x) = \mathcal{E} \cup \Big\{ i \in \mathcal{I} \;\Big|\; c_i(x) = 0 \Big\}. \tag{2.9}$$

Tout dépendant de la nature de $f(x)$, il est possible qu'une solution de (2.7) soit un *minimum global*, ou encore, un *minimum local*.

**Définition 1** *On dit qu'un point $x^\star \in \Omega$ est un minimum global si $f(x^\star) \leq f(x)\ \forall\ x \in \Omega$.*

**Définition 2** *On dit qu'un point $x^\star \in \Omega$ est un minimum local s'il existe un voisinage $\mathcal{N}$ de $x^\star$ tel que $f(x^\star) \leq f(x)\ \forall\ x \in \mathcal{N} \cap \Omega$.*

D'autre part, afin de satisfaire les exigences des conditions d'optimalité *de premier ordre*, il est nécessaire de supposer — dans (2.7) — que $f(x),\ c_i(x) : \mathbb{R}^n \to \mathbb{R} \in \mathcal{C}^1\ \forall\ i \in \mathcal{E} \cup \mathcal{I}$.

Avant de présenter les conditions d'optimalité d'ordre un, nous détaillons la condition de *qualification d'indépendance linéaire des contraintes*, ou *linear independance constraint qualification* (LICQ).

**Définition 3** *On dit que la LICQ est satisfaite en $x^\star \in \Omega$ si les gradients des contraintes actives en $x^\star$ sont linéairement indépendants, i.e.,*

$$\Big\{ \nabla c_i(x),\ \forall\ i \in \mathcal{A}(x^\star) \Big\} \textit{ forme un ensemble linéairement indépendant.} \tag{2.10}$$

Nous définissons également le Lagrangien du problème (2.7) :

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x), \tag{2.11}$$

où $\lambda_i$ est le multiplicateur de Lagrange associé à la $i^{\text{ème}}$ contrainte.

Nous pouvons maintenant définir les conditions de $\mathcal{KKT}$ pour (2.7). Ces conditions sont également valides pour un problème sans contraintes.

**Théorème 1** *Soit $x^\star$ un minimum local de* (2.7) *où la condition LICQ est respectée et $f,\ c_i \in \mathcal{C}^1\ \forall\ i \in \mathcal{E} \cup \mathcal{I}$.*

*Alors, il existe un unique vecteur $\lambda^\star$ tel que les conditions suivantes sont satisfaites à $(x^\star, \lambda^\star)$ :*

$$\nabla_x \mathcal{L}(x^\star, \lambda^\star) = 0, \tag{2.12}$$
$$c_i(x^\star) = 0, \qquad \forall\ i \in \mathcal{E}, \tag{2.13}$$
$$c_i(x^\star) \geq 0, \qquad \forall\ i \in \mathcal{I}, \tag{2.14}$$
$$\lambda_i^\star \geq 0, \qquad \forall\ i \in \mathcal{I}, \tag{2.15}$$
$$\lambda_i^\star c_i(x^\star) = 0, \qquad \forall\ i \in \mathcal{I}\,. \tag{2.16}$$

*Ces conditions sont nommées les conditions d'optimalité d'ordre un, car elles utilisent principalement l'information du gradient. Elles sont couramment nommées les conditions de $\mathcal{KKT}$, pour Karush-Kuhn-Tucker. Si la condition LICQ n'est pas respectée, alors il pourrait exister plusieurs vecteurs $\lambda^\star$, ou aucun, tel que les conditions de $\mathcal{KKT}$ sont satisfaites.*

On dit que $x^\star \in \Omega$ est un point stationnaire s'il existe un vecteur $\lambda^\star$ tel que (2.12)- (2.16) sont vérifiées.

Certains problèmes, tel (SP), ont la particularité d'être *convexes*, ce qui garantit que $x^\star$ est un minimum global.

En théorie, des conditions d'ordre deux sont également nécessaires pour déterminer la nature d'un point stationnaire, mais elles sont rarement utilisées dans la mise en œuvre des algorithmes. D'autre part, nous considérons spécifiquement un problème convexe sous contraintes linéaires, de sorte que les conditions d'ordre deux sont satisfaites d'emblée. Nous laissons le lecteur se référer à (Nocedal et Wright, 2000, section 12.5) pour s'en convaincre.

## 2.3 Méthodes de contraintes actives à base de projections

Cette section propose un résumé des méthodes de contraintes actives, qui sont au coeur de nos travaux. À cette fin, nous introduisons le problème générique :

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.c.} \quad & c_i^\mathsf{T} x \geq 0, \quad i \in \mathcal{I}, \end{aligned} \tag{2.17}$$

qui est de la forme du problème (SP) que nous cherchons à résoudre et où $c_i \in \mathbb{R}^n$. Précisons que les contraintes de bornes sont un cas particulier des inégalités linéaires, de sorte que les concepts exposés dans cette section s'y appliquent également.

Plus particulièrement, nous introduisons le concept de *projection* dans l'ensemble admissible et détaillons une méthode de contraintes actives à base de projections, soit TRON. TRON est

une méthode de Newton projetée sous région de confiance développée par Lin et Moré (1999). Nous portons une attention particulière à ce solveur, car nous avons été amenés à l'adapter au problème (SP), soit au cas de contraintes d'inégalité linéaires. Bien que compatible avec les problèmes de la forme (2.17), en pratique, TRON est principalement employé sur des problèmes de bornes pour des raisons qui seront mises en évidence aux sections 2.3.1, 2.3.2 et 2.3.3. L'absence de littérature claire sur le sujet, combinée à sa prépondérance au sein de nos travaux, motive une présentation plus minutieuse de l'algorithme. Avant tout, nous discutons du fonctionnement d'une méthode de contraintes actives générique.

### 2.3.1 Méthodes de contraintes actives

Soit $x^\star$ un minimum local du problème (2.17). Les méthodes de contraintes actives sont basées sur la constatation suivante : si l'ensemble de contraintes actives $\mathcal{A}(x^\star)$ (2.9) était connu d'avance, alors on pourrait obtenir $x^\star$ en résolvant le problème sous contraintes d'égalité

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.c.} \quad & c_i^\mathsf{T} x = 0, \quad i \in \mathcal{A}(x^\star). \end{aligned} \tag{2.18}$$

Le problème (2.18) correspond à minimiser $f(x)$ sur la *face active* en $x^\star$. La validité de ce résultat peut être confirmée par observation des conditions de $\mathcal{KKT}$ (2.12), (2.14) et (2.16) du problème général (2.7) : pour toute contrainte $c_i(x)$ *inactive*, $c_i(x) > 0$ et son multiplicateur $\lambda_i = 0$.

Pour un itéré $x_k$, les méthodes de contraintes actives approximent $\mathcal{A}(x_k) \approx \mathcal{A}(x^\star)$ et résolvent (2.18). L'ensemble actif est ensuite mis à jour en fonction des contraintes violées, du gradient de la fonction objectif et des multiplicateurs de Lagrange. L'ajout de contrainte à l'ensemble actif se fait généralement à raison d'*une seule par itération*. Puisqu'il y a un nombre fini d'ensembles actifs possibles pour un ensemble de contraintes donné, la convergence de l'algorithme en un nombre fini d'itérations est assurée *en arithmétique exacte*. Néanmoins, cela soulève un problème de nature combinatoire : le nombre d'ensembles actifs à tester grandit exponentiellement avec le nombre de contraintes du problème. Dans ce contexte, l'utilisation d'un *pas projeté* permet d'inclure ou d'exclure plusieurs contraintes de l'ensemble actif par itération, de sorte que le nombre d'itérations requis pour atteindre une solution est diminué drastiquement. Pour plus d'information sur les méthodes de contraintes actives en général, le lecteur peut se rapporter à, e.g., Luenberger et Ye (2008).

### 2.3.2 Projections sur l'ensemble admissible

Les méthodes projetées sont particulièrement appropriées aux problèmes d'optimisation de la forme suivante,

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.c.} \quad & x \in \mathcal{S}, \end{aligned} \tag{2.19}$$

où $\mathcal{S}$ est un ensemble convexe fermé. Ces deux propriétés supplémentaires sur $\mathcal{S}$ permettent d'assurer que la projection $\mathcal{P}_{\mathcal{S}}$ d'un point arbitraire $\bar{x}$ sur $\mathcal{S}$ est bien définie et unique — voir, e.g., (Boyd et Vandenberghe, 2010). En outre, il est possible d'obtenir ce point en résolvant le problème d'optimisation convexe :

$$\begin{aligned} \min_x \quad & \tfrac{1}{2}\left\|x - \bar{x}\right\|^2 \\ \text{s.c.} \quad & x \in \mathcal{S}. \end{aligned} \tag{2.20}$$

Le problème (2.20) illustre une difficulté potentielle liée aux méthodes de projection en général : (2.20) peut être aussi coûteux à résoudre que le problème original (2.19). Pour cette raison, peu d'attention est portée aux méthodes projetées, hormis dans les cas où il est aisé de projeter sur l'ensemble de contraintes, i.e. il existe une solution analytique à (2.20).

#### Projection sur contraintes de bornes

Le cas de figure le plus simple est celui de bornes :

$$\mathcal{B} = \left\{ x \;\middle|\; l \leq x \leq u \right\}, \tag{2.21}$$

pour lequel la projection de $\bar{x}$ est donnée explicitement par :

$$\mathcal{P}_{\mathcal{B}}[\bar{x}] = \min(\max(\bar{x},\ l),\ u), \tag{2.22}$$

où les opérations sont effectuées terme à terme. Le lecteur peut se rapporter à, e.g, (Boyd et Vandenberghe, 2010) pour une dérivation complète de ce résultat. Plusieurs solveurs pour problèmes bornés, dont TRON, cherchent à tirer profit du faible coût de cette opération de projection.

### 2.3.3 TRON : méthode de Newton avec région de confiance

Le solveur TRON est une méthode de Newton avec région de confiance qui regroupe une panoplie des mécanismes parmi les plus importants de l'optimisation : la construction d'ap-

proximations quadratiques de la fonction objectif, le gradient conjugué, les projections sur l'ensemble admissible, la direction de descente de Newton et l'utilisation d'un ensemble actif ainsi que d'une région de confiance.

Puisque la généralisation de TRON au cas d'inégalités linéaires est présentée au chapitre 4, nous limitons pour le moment notre analyse au problème avec contraintes de bornes

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.c.} \quad & l \leq x \leq u. \end{aligned} \tag{2.23}$$

C'est d'ailleurs l'implémentation la plus commune de TRON, principalement à cause du faible coût de la projection sur les bornes (2.22). Nous utilisons la même notation qu'à la section 2.3.2 pour désigner les contraintes de bornes (2.21). TRON procède en minimisant itérativement l'approximation quadratique de la fonction objectif

$$\psi(w) = g_k^{\mathsf{T}} w + \tfrac{1}{2} w^{\mathsf{T}} H_k w, \tag{2.24}$$

où $x_k$ est un itéré, $g_k = \nabla f(x_k)$ et $H_k = \nabla^2 f(x_k)$. Cette idée s'apparente fortement aux méthodes SQP et le lecteur peut se référer à, e.g., (Nocedal et Wright, 2000, chapitre 18) pour plus d'information sur ces dernières. L'utilisation additionnelle d'une région de confiance, notée $\Delta_k$, permet d'assurer que les itérés générés demeurent dans un voisinage où $\psi(w)$ représente bien la fonction $f(x_k + w)$. Le lecteur peut se référer au livre de Conn *et al.* (2000) pour plus d'information sur l'utilisation de régions de confiance dans le domaine de l'optimisation.

Avant de discuter du coeur de l'algorithme, nous introduisons le concept des *points de cassures* ainsi que le *pas* et le *point de Cauchy*. Ces mécanismes seront utilisés ultérieurement dans l'algorithme de TRON.

### Points de cassure

Les *points de cassure* (*break points*) correspondent aux longueurs de pas que l'on peut prendre selon une direction $w$ depuis un itéré $x_k$, avant qu'une contrainte *inactive* devienne *active*. Pour ce faire, on définit avant tout les ensembles $\mathscr{L}$ et $\mathscr{U}$ représentant les contraintes susceptibles de devenir actives respectivement pour les bornes inférieures et supérieures :

$$\begin{aligned} \mathscr{L} &= \Big\{ i \;\Big|\; x_{ki} - l_i > 0 \;\wedge\; w_i < 0 \Big\}, \\ \mathscr{U} &= \Big\{ i \;\Big|\; u_i - x_{ki} > 0 \;\wedge\; w_i > 0 \Big\}, \end{aligned} \tag{2.25}$$

où $x_{ki}$ désigne la $i^{\text{ème}}$ composante de $x_k$ et de même pour les autres vecteurs. Connaissant ces ensembles d'indices, Lin et Moré (1999) obtiennent $\alpha_{\mathscr{L}} \geq 0$ et $\alpha_{\mathscr{U}} \geq 0$, les longueurs de pas associées respectivement aux bornes inférieures et supérieures, en calculant

$$\begin{aligned} \alpha_{\mathscr{L}i} &= \frac{l_i - x_i}{w_i} \quad \forall\ i \in \mathscr{L}, \\ \alpha_{\mathscr{U}i} &= \frac{u_i - x_i}{w_i} \quad \forall\ i \in \mathscr{U}, \end{aligned} \tag{2.26}$$

où $\alpha_{\mathscr{L}i}$ représente la $i^{\text{ème}}$ composante de $\alpha_{\mathscr{L}}$, pareillement pour $\alpha_{\mathscr{U}i}$. Dans TRON, les points de cassure sont principalement utilisés pour borner les recherches linéaires projetées, comme il sera démontré ultérieurement. En conséquence, nous nous intéressons spécifiquement aux longueurs de pas maximales et minimales :

$$\begin{aligned} \alpha_{\min} &= \max\{\alpha_{\mathscr{L}},\ \alpha_{\mathscr{U}}\}, \\ \alpha_{\max} &= \min\{\alpha_{\mathscr{L}},\ \alpha_{\mathscr{U}}\}, \end{aligned} \tag{2.27}$$

où les opérations sont effectuées terme à terme.

### Point et pas de Cauchy

Afin d'assurer la convergence de l'algorithme, Lin et Moré (1999) ont recours au *pas de Cauchy* [1]. Ce pas de Cauchy est défini selon $s^C = s(\hat{\alpha})$, où

$$s(\alpha) = \mathcal{P}_{\mathcal{B}}\left[x_k - \alpha g_k\right] - x_k, \tag{2.28}$$

et $\mathcal{P}_{\mathcal{B}}$ est l'opération de projection définie à l'équation (2.22). L'équation (2.28) correspond en fait à une direction de descente du *gradient projeté*, pour laquelle nous imposons $\hat{\alpha}$ tel que les conditions

$$\begin{aligned} \psi(s(\hat{\alpha})) &\leq \mu_0 g_k^{\mathsf{T}} s(\hat{\alpha}), \\ \left\|s(\hat{\alpha})\right\| &\leq \mu_1 \Delta_k, \\ \hat{\alpha} &\geq 0, \end{aligned} \tag{2.29}$$

sont respectées, où $\mu_0 > 0$ et $\mu_1 > 0$ sont des paramètres permettant respectivement de pondérer la décroissance de la quadratique et le rayon de la région de confiance. La première équation impose que le pas de Cauchy calculé introduise une décroissance suffisante de l'approximation quadratique, tandis que la deuxième équation impose de calculer un pas de Cauchy qui respecte la borne de la région de confiance.

1. Ce pas de Cauchy est en fait le pas de Cauchy *généralisé* aux contraintes linéaires. Pour plus de détails, le lecteur peut se référer à (Conn *et al.*, 2000, chapitre 12.2).

En pratique, $\hat{\alpha}$ est obtenu en interpolant sur $\alpha$, i.e. en posant $\alpha \leftarrow \tau\alpha$ avec $\tau \in ]0,1[$, jusqu'à ce que les conditions (2.29) soient satisfaites. Si les conditions (2.29) sont satisfaites d'emblée, Lin et Moré (1999) proposent d'extrapoler sur $\alpha$, i.e. en posant $\alpha \leftarrow \tau\alpha$ avec $\tau > 1$, et en itérant jusqu'à ce que (2.29) soient violées ou que $\alpha > \alpha_{\max}$, avec $\alpha_{\max}$ définit selon (2.27). En effet, si $\alpha$ excède le *point de cassure maximal*, le résultat de la projection dans (2.28) devient invariant. À ce moment, on peut récupérer le dernier $\hat{\alpha}$ acceptable et quitter la procédure. L'objectif de l'extrapolation est d'amener l'algorithme à prendre de plus grands pas, de sorte à accélérer la décroissance de la fonction objectif. Finalement, le *point de Cauchy* $x_k^C$ peut-être obtenu depuis un pas de Cauchy selon $x_k^C = x_k + s^C$.

Cela nous mène à l'étape suivante, qui est le coeur de l'algorithme, et qui consiste à obtenir un pas $s$ en résolvant le sous-problème de région de confiance. Afin d'assurer la convergence de la méthode, Lin et Moré (1999) imposent que $s$ respecte les conditions

$$\begin{aligned} \psi(s) &\leq \mu_0 \psi(s^C), \\ \left\| s \right\| &\leq \mu_1 \Delta_k, \\ x_k + s &\in \mathcal{B}. \end{aligned} \tag{2.30}$$

Les deux dernières conditions permettent de garantir le respect de la région de confiance et de l'ensemble admissible, tandis que la première assure la convergence de l'algorithme en bornant la décroissance par celle obtenue avec le pas de Cauchy. Il est important de noter que les conditions (2.30) peuvent toujours être satisfaites en prenant $s = s^C$ par définition. Par contre, ce choix revient à réduire la méthode à une descente de gradient projeté, ce qui risque d'entraîner une convergence lente.

### Sous-problème de région de confiance

Tout comme dans le cadre des méthodes de contraintes actives présentées à la section 2.3.1, on cherche à minimiser l'approximation quadratique sur la *face active* des contraintes, tout en respectant la région de confiance. Or, Lin et Moré (1999) procèdent de manière astucieuse : en minimisant itérativement $\psi(w)$ sur la face active, on peut non seulement mettre à jour l'ensemble actif au fil des itérations, mais aussi prendre des pas projetés. Cela favorise l'ajout de contraintes à l'ensemble actif, de sorte qu'il est possible d'identifier l'ensemble actif optimal plus rapidement.

Pour ce faire, Lin et Moré (1999) génèrent une suite d'*itérés mineurs* $x_{k,j}$ obtenus depuis un minimum approché $w^\star$ de $\psi(w)$, où les indices $k, j$ indiquent le $j^{\text{ème}}$ itéré mineur depuis $x_k$. Bien que ces concepts peuvent sembler ambigus pour le lecteur, il s'agit de l'intuition derrière

la procédure qui est au coeur de TRON et nous les formalisons mathématiquement au cours de cette section.

Le premier itéré mineur est choisi comme étant le point de Cauchy calculé précédemment, i.e. $x_{k,1} = x_k^C$. Lin et Moré (1999) imposent les conditions suivantes sur les itérés mineurs :

$$\begin{aligned} x_{k,j} &\in \mathcal{B}, \\ \mathcal{A}(x_k^C) &\subset \mathcal{A}(x_{k,j}), \\ \left\| x_{k,j} - x_k \right\| &\leq \mu_1 \Delta_k. \end{aligned} \tag{2.31}$$

La première et la troisième condition assurent que les itérés respectent les contraintes de bornes et la région de confiance. La deuxième condition est nécessaire pour garantir la convergence de la méthode : on impose que l'ensemble actif à $x_{k,1} = x_k^C$ soit un sous-ensemble de tous les prochains ensembles actifs, i.e. toutes les contraintes actives à $x_{k,j}$ doivent demeurer actives pour tout $x_{k,l}$ avec $l \geq j$.

Rappelons que le prochain itéré mineur $x_{k,j+1}$ est obtenu à partir d'une direction de descente depuis $x_{k,j}$, notée $w^\star$. Cette direction est obtenue en résolvant le sous-problème de région de confiance

$$\begin{aligned} \min_w \quad & \psi(w + x_{k,j} - x_k) \\ \text{s.c.} \quad & w_i = 0, \ i \in \mathcal{A}(x_{k,j}) \\ & \left\| w + x_{k,j} - x_k \right\| \leq \Delta_k, \end{aligned} \tag{2.32}$$

de sorte à calculer $w^\star$ depuis $x_{k,j}$, mais relativement à $x_k$. Par observation de (2.32), on peut remarquer que demeurer sur la face active $w_i = 0, \ i \in \mathcal{A}(x_{k,j})$ revient à traiter le sous-problème *réduit*, c'est-à-dire à minimiser uniquement sur les *variables libres*. Ce sous-problème peut s'écrire :

$$\begin{aligned} \min_v \quad & (Bg_k)^\mathsf{T} v + \tfrac{1}{2} v^\mathsf{T} B H_k B^\mathsf{T} v \\ \text{s.c.} \quad & \left\| v \right\| \leq \Delta_k, \end{aligned} \tag{2.33}$$

où on a posé $v = B(w + x_{k,j} - x_k)$, avec $B$ la matrice de *restriction* composée des lignes de l'identité correspondant à $i \notin \mathcal{A}(x_{k,j})$. En pratique, Lin et Moré (1999) appliquent la procédure de gradient conjugué tronqué de Steihaug (1983) au problème (2.33) afin d'obtenir un minimum approché $v^\star$, qui correspond aux composantes non-nulles de $w^\star$.

On *explore* ensuite la face active en effectuant une recherche linéaire projetée, depuis $x_{k,j}$ et dans la direction $w^\star$ :

$$x_{k,j+1} = \mathcal{P}_\mathcal{B}\left[ x_{k,j} + \alpha w^\star \right]. \tag{2.34}$$

Le pas $\alpha$ est obtenu par interpolation, i.e. $\alpha \leftarrow \tau\alpha$ avec $\tau \in ]0, \ 1[$, tel que le critère de

décroissance sur l'approximation quadratique suivant soit respecté :

$$\psi(x_{k,j+1} - x_k) \leq \psi(x_{k,j} - x_k) + \mu_0 \min\{\nabla\psi(x_{k,j} - x_k)^\mathsf{T}(x_{k,j+1} - x_{k,j}),\ 0\}, \tag{2.35}$$

ou encore que $\alpha \leq \alpha_{\min}$, où $\alpha_{\min}$ est défini selon (2.27). Le point $x_{k,j+1}$ obtenu correspond au prochain itéré mineur. La procédure précédente a pour but d'identifier si le minimum global de $\psi$ est sur la face active courante, car elle *incite* les contraintes inactives à entrer dans l'ensemble actif. On tente de trouver un $\alpha > \alpha_{\min}$ afin d'effectuer de plus grands pas et de faire entrer plusieurs contraintes dans l'ensemble actif. Par (2.27) et (2.34), on peut conclure que choisir $x_{k,j+1}$ tel que $\alpha = \alpha_{\min}$ garantit de faire entrer *une* contrainte dans l'ensemble actif, de sorte que le nombre d'itérations requis pour résoudre (2.32) est borné par le nombre de contraintes.

Lin et Moré (1999) identifient un minimum de (2.32) à $x_{k,j}$ donné grâce à la mesure d'optimalité :

$$\Big\|B\nabla\psi(x_{k,j} - x_k)\Big\| \leq \epsilon\Big\|Bg_k\Big\|, \tag{2.36}$$

où $\epsilon$ est une tolérance. Si (2.36) est satisfaite, ou encore que le pas $x_{k,j} - x_k$ atteint la borne de la région de confiance, i.e. $\Big\|x_{k,j} - x_k\Big\| > \Delta_k$, on peut quitter la procédure et récupérer le dernier point acceptable. Nous devons maintenant confirmer que $\psi$ est une représentation adéquate de $f$ en $x_{k,j}$ avant d'accepter le nouvel itéré.

### Mise à jour de la région de confiance et acceptation du pas

Soit $x_{k,j}$ le dernier itéré mineur obtenu par la procédure expliquée à la section 2.3.3. On peut définir le ratio $\rho$ entre la réduction prédite et la réduction actuelle

$$\rho = \frac{f(x_{k,j}) - f(x_k)}{\psi(x_{k,j} - x_k)} \tag{2.37}$$

afin d'obtenir une mesure additionnelle de la précision de notre modèle quadratique par rapport à $f(x)$. Si le ratio est en deçà d'un seuil $\eta_0 \in ]0,\ 1[$, on juge que $\psi(s)$ n'était pas une représentation adéquate de la fonction objectif et on rejette le pas. Mathématiquement, on peut formuler la règle de décision suivante :

$$x_{k+1} = \begin{cases} x_k + s, & \text{si } \rho > \eta_0, \\ x_k, & \text{si } \rho \leq \eta_0. \end{cases} \tag{2.38}$$

Une fois le ratio (2.37) calculé, Lin et Moré (1999) proposent de mettre à jour la région de confiance $\Delta_k$ à l'aide de deux séries de paramètres. La première implique les constantes $0 < \eta_1 < \eta_2 < 1$ et permet de déterminer quelle règle utiliser pour mettre à jour $\Delta_k$. La seconde implique les constantes $0 < \sigma_1 < \sigma_2 < 1 < \sigma_3$ et permet de pondérer la mise à jour de $\Delta_k$. Les règles définies par Lin et Moré (1999) sont les suivantes :

$$\begin{aligned} &\Delta_{k+1} \in \left[\sigma_1 \min\left\{\left\|s\right\|,\ \Delta_k\right\},\ \sigma_2\Delta_k\right], && \text{si } \rho \leq \eta_1, \\ &\Delta_{k+1} \in \left[\sigma_1\Delta_k,\ \sigma_3\Delta_k\right], && \text{si } \rho \in (\eta_1,\ \eta_2), \\ &\Delta_{k+1} \in \left[\Delta_k,\ \sigma_3\Delta_k\right], && \text{si } \rho \geq \eta_2. \end{aligned} \tag{2.39}$$

### Algorithme TRON

Nous avons maintenant tous les outils requis afin de présenter TRON. L'algorithme 4 résume notre implémentation.

**Algorithme 4** Algorithme TRON

1: Initialiser $x_0 = \mathcal{P}_{\mathcal{B}}\left[x_0\right]$, évaluer $f_0 = f(x_0)$, $g_0 = \nabla f(x_0)$ et $H_0 = \nabla^2 f(x_0)$
2: **tant que** critères d'arrêt non respectés **faire**
3: Calculer le pas de Cauchy (2.28) qui respecte (2.29).
4: **tant que** $\left\|s\right\| \leq \Delta_k \ \wedge\ \left\|B\nabla\psi(s)\right\| > \epsilon\left\|Bg_k\right\|$ **faire**
5: Mettre à jour $\mathcal{A}(x_{k,j})$ et $B$.
6: Résoudre le sous-problème de région de confiance (2.33).
7: Calculer un itéré mineur (2.34) qui respecte (2.35) et (2.31).
8: **fin tant que**
9: Appliquer les règles (2.38) et (2.39) au dernier point $x_{k,j}$.
10: **fin tant que**

# CHAPITRE 3 DÉMARCHE DE L'ENSEMBLE DU TRAVAIL

Tel que mentionné à la section 1.3, nous cherchons à appliquer des méthodes d'ensemble actif à base de projections au problème (SP). Notre intérêt envers les méthodes projetées provient d'une hypothèse que nous avons posée, à savoir qu'il est possible de concevoir des projections *efficaces* sur l'ensemble réalisable du problème (SP). Cette hypothèse est motivée par le faible coût des produits avec la matrice de mise à l'échelle et la possibilité de traiter le *dual* du problème de projection, qui est considérablement plus facile à résoudre. Ce sujet est discuté de manière exhaustive dans notre article à la section 4.

Nous discutons des méthodes de contraintes actives dans un cadre général à la section 2.3.1 et en détaillons une variante à base de projections à la section 2.3.3. Nous portons une attention particulière à TRON, car, dans l'article, nous insistons uniquement sur les modifications apportées. En premier lieu, notre intérêt envers TRON provient du fait que nous cherchons à opposer des méthodes projetées d'ordre un et d'ordre deux sur le problème (SP). Étant donné le nombre relativement élevé de variables, nous avons raison de croire que l'une pourrait bénéficier des désavantages de l'autre. La méthode d'ordre un que nous sommes amenés à étudier est la méthode du gradient projeté spectral, une variante de la descente de gradient pour problèmes sous contraintes.

Notre article propose une comparaison rigoureuse de ces deux méthodes, en plus d'une description détaillée des opérations de projection qui doivent être développées afin de pouvoir les appliquer. Nous évaluons leur performance respective sur des données synthétiques et les comparons au solveur L-BFGS-B sur le problème (OP) en coordonnées cartésiennes, tel que recommandé par Hamelin (2009). Ultimement, nous souhaitons bénéficier des gains en mémoire apportés par la discrétisation en coordonnées cylindriques, tout en demeurant compétitif avec L-BFGS-B appliqué au problème en coordonnées cartésiennes.

# CHAPITRE 4 ARTICLE 1: FACTORIZATION-FREE METHODS FOR COMPUTED TOMOGRAPHY

## Factorization-free Methods for Computed Tomography

by Yves Goussard, Maxime McLaughlin and Dominique Orban

Manuscript submitted to *SIAM Journal on Imaging Sciences* (SIIMS).

**Abstract**

We study X-ray tomographic reconstruction using statistical methods. The problem is expressed in cylindrical coordinates, which yield significant computational and memory savings, with nonnegativity bounds. A change of variables involving a Fourier matrix attempts to improve the conditioning of the Hessian but introduces linear inequality constraints. The scale and density of the problem call for factorization-free methods. We argue that projections into the feasible set can be computed efficiently by solving a bound-constrained linear least-squares problem with a fast operator. This motivates our interest towards projection-based active-set methods for the reconstruction problem, namely a spectral projected gradient method and a trust-region projected Newton method that we generalize to our specific scenario. For the projection subproblem, we consider several projection-based methods for bound-constrained problems. We assess the performance of several algorithm combinations on the reconstruction problem using synthetic data. Our results show that the projected Newton method combined with efficient projection strategies applied to the problem in cylindrical coordinates with linear inequality constraints is competitive in terms of run time with a limited-memory BFGS applied to the problem in cartesian coordinates with simple bounds.

**Keywords** factorization-free, convex constrained optimization, tomographic reconstruction, medical imaging

**AMS subject classifications** 49M15, 49M29, 90C25, 90C90, 92C55, 94A08

### 4.1 Introduction

Generally speaking, tomographic reconstruction methods fall within two categories: *analytical* techniques, which rely on strong approximations to the data formation model, and *statistical* or *algebraic* techniques, which make use of an estimation methodology. Since the inception of X-ray tomography, analytical methods have been prevalent in clinical settings, mainly because of their limited computational requirements. However, their limitations in

terms of accuracy has been recognized early (Herman et Rowland, 1973), and statistical methods have been repeatedly shown to produce superior results in many respects (Pan *et al.*, 2009; Beister *et al.*, 2012). Nevertheless, practical use of statistical methods has remained limited, mostly because of high computation times and large memory requirements. Recently, statistical approaches have been the subject of renewed interest due to the increase in computer performance, pressure toward X-ray dose reduction and development of new types of X-ray scanners, even though computation time and memory footprint remain major difficulties.

An avenue to tackle these difficulties is to make use of nonstandard representations that can take advantage of redundancies in the data collection process. Among them, formulation of the problem in cylindrical coordinates (Thibaudeau *et al.*, 2013; Goussard *et al.*, 2013) has been shown to produce considerable reduction of memory footprint without on the fly computation of the projection matrix, and significant acceleration of the computation through straightforward parallelization. However, expressing the reconstruction problem in cylindrical coordinates induces substantial ill conditioning, but the latter can be alleviated by appropriate scaling. The main focus of our work is to investigate the various algorithms that can solve the scaled problem and account for the nonnegativity constraints that the solution must satisfy.

Our paper is organized as follows. In section 4.2, we describe how the reconstruction problem is obtained from maximum a posteriori estimation. Its key features are that it is a large-scale regularized linear least-squares problem with linear inequality constraints. To tackle this optimization problem, we consider projection-based factorization-free active-set methods. We outline a generic active-set method as well as the various projection operations that are required by our algorithms in section 4.3 and detail the methods that we use in section 4.4. Because some of the projection operations are cast as other optimization problems, we address their solution in section 4.5. We report the results of our reconstruction algorithm on synthetic data in section 4.6, and we study its behavior according to the choice of reconstruction solver, projection solver and various parameters. Conclusions, as well as further improvements, appear in section 4.7.

Implementations of our solvers are available in object-oriented MATLAB as part of the NLPLab optimization framework available at `https://bitbucket.org/maxmcl/nlplab`.

## 4.2 Iterative Reconstruction Algorithm

X-ray tomography is an imaging modality based on the measurement of X-ray attenuation trough an unknown object under several incidences. The goal of reconstruction is to recover the spatial distribution of linear X-ray attenuation coefficients in the object from the measurements and a model of the X-ray attenuation process—the data formation model. Typically, such models are based upon the Beer-Lambert law. Here, we consider the *stochastic* version of the Beer-Lambert law, and we show how probability estimation can be used to perform the reconstruction.

### 4.2.1 Stochastic Beer-Lambert Law and Discretization

In tomographic reconstruction, the Beer-Lambert law relates the attenuation of an energy beam, that travels through an object, with a distribution of attenuation coefficients $\mu(x) : \mathbb{R}^{n_{\text{dim}}} \to \mathbb{R}$, where $n_{\text{dim}}$ is the number of spatial dimensions (typically 1, 2 or 3). We assume that the uncertainty on the transmitted intensities is dominated by the quantum effects related to the attenuation of the X-ray beams by matter. Under that hypothesis, the photon counts collected by the detectors can be modeled as a Poisson distribution $\mathcal{P}(l)$, where $l$ is the parameter of the distribution. We introduce the random variable $N$ representing the photon counts of realization vector $n \in \mathbb{R}^{n_{\text{meas}}}$, where $n_{\text{meas}}$ is the number of intensity measurements. Furthermore, we assume both the attenuation coefficients and the source to be energy *independent*. Consequently, the *monochromatic* stochastic Beer-Lambert law that will be considered hereafter is

$$N \sim \mathcal{P}\left(n_0 \, \mathrm{e}^{-\int_{L_i} \mu(x) \, \mathrm{d}x}\right), \tag{4.1}$$

where $n_0 \in \mathbb{R}$ is the peak energy of the source, $L_i$ represents a linear path through the patient $\mu(x)$ and $n_i$ an intensity measurement collected by a detector.

The *sinogram*, often called *projections* (not to be confused with mathematical projections) or *projection data*, is

$$y := \ln\left(\frac{n_0}{n}\right), \tag{4.2}$$

where the logarithm and division occur componentwise.

In practice, $n_{\text{meas}}$ is determined by the angular discretization of the rotation of the source and the detectors, as well as the total number of detectors. Assuming three-dimensional (3D) reconstruction, the subscript $i = 1, \ldots, n_{\text{meas}}$ denotes an $i$-th measurement, obtained at an $i$-th scan angle, detector and axial position.

In order to obtain an expression of (4.1) that is suited to numerical methods, the domain of $\mu$ must be discretized in $n_{\text{vox}}$ voxels, such that $\mu_j$, the $j$-th component of $\mu$, is assigned to the $j$-th voxel. This is done by the means of a discretization function $\xi(x)$ that can be interpreted as an $n_{\text{dim}}$-dimensional mesh. Hence, $\mu$ becomes independent of $x$ and can be excluded from the integrand of (4.1), which becomes a collection of ray-voxel intersection lengths along $L_i$,

$$\int_{L_i} \mu(x)\,\mathrm{d}x \int_{L_i} \sum_j \mu_j \xi_j(x)\,\mathrm{d}x = \sum_j \mu_j \int_{L_i} \xi_j(x)\,\mathrm{d}x = \sum_j p_{ij}\mu_j,$$

where we define

$$p_{ij} := \int_{L_i} \xi_j(x)\,\mathrm{d}x.$$

We call $\mu \in \mathbb{R}^{n_{\text{vox}}}$ the vector of attenuation coefficients, and $P \in \mathbb{R}^{n_{\text{meas}} \times n_{\text{vox}}}$ the *projection matrix* that contains the intersection lengths. We may now rewrite (4.1) as

$$N \sim \mathcal{P}\left(n_0\,\mathrm{e}^{-P\mu}\right). \tag{4.3}$$

Assuming independent and identically distributed (i.i.d.) photon counts, (4.3) has the conditional probability density function

$$P(N = n \mid \mu) = \prod_i \left[\frac{\exp\left(-n_0\,\mathrm{e}^{-[P\mu]_i}\right)\left(n_0\,\mathrm{e}^{-[P\mu]_i}\right)^{n_i}}{n_i!}\right], \tag{4.4}$$

where $[P\mu]_i$ designates the $i$-th component of $P\mu$. Hence, using this expression, we can evaluate the probability of measuring the intensities $n$ given $\mu$. Naturally, we will seek the distribution that is the *most likely* according to our data, which corresponds to *maximum likelihood* estimation.

### 4.2.2 Maximum Likelihood

The *maximum likelihood* (ML) estimator of the conditional probability distribution (4.4) is

$$\hat{\mu}_{\text{ML}} = \operatorname{argmax}\ P(N = n \mid \mu)\ \text{ subject to } \mu \geq 0, \tag{4.5}$$

where we impose the physical constraint $\mu \geq 0$. Instead of maximizing (4.4), we may equivalently *minimize* the negative log-likelihood,

$$L(n \mid \mu) = \sum_i \left[n_0\,\mathrm{e}^{-[P\mu]_i} + n_i[P\mu]_i + \log(n_i!)\right]. \tag{4.6}$$

Minimizing (4.6) under nonnegativity constraints remains difficult given its nonlinear nature and compels us to consider different approaches. Sauer et Bouman (1993) circumvent this issue by applying a second-order Taylor expansion to (4.6), where each term is expanded about $y$, so that, after dropping terms that do not depend on $\mu$, the objective function reduces to a least-squares residual

$$L(n \mid \mu) \approx \tfrac{1}{2}\left\|P\mu - y\right\|_{\Delta_N}^2 \tag{4.7}$$

where $y$ is defined in (4.2) and $\Delta_N = \operatorname{diag}(n_i)_{i=1,\,\ldots,\,n_{\text{meas}}}$ acts as a weighing matrix, where greater penalties are assigned to higher values of $n_i$. Indeed, since they correspond to less attenuated beams, they have higher signal-to-noise ratio and thus less uncertainty. To further simplify our model, we set $\Delta_N = I$. Finally, by substituting (4.7) into (4.5), the maximum likelihood estimator becomes:

$$\hat{\mu}_{\text{ML}} = \operatorname{argmin} \tfrac{1}{2}\left\|P\mu - y\right\|^2 \text{ subject to } \mu \geq 0. \tag{4.8}$$

One might note that (4.8) can also be obtained by assuming that the measured photon counts follow a normal distribution. Moreover, since we know that $\mu$ describes biological tissues, we might expect the reconstructed $\mu$ to follow certain behaviors. Exploiting this *prior knowledge* is possible through *maximum a posteriori* estimation.

### 4.2.3 Maximum A Posteriori and Penalty Function

Bayes's theorem lets us introduce a prior distribution $P(\mu)$ that reflects our knowledge on $\mu$. The *maximum a posteriori* (MAP) estimate is the most probable value of $P(\mu \mid n)$ taking the physical constraint $\mu \geq 0$ into account, i.e.,

$$\hat{\mu}_{\text{MAP}} \in \operatorname{argmax} \frac{P(n \mid \mu)\, P(\mu)}{P(n)} \text{ subject to } \mu \geq 0. \tag{4.9}$$

Instead of maximizing (4.9), we once again minimize the negative of its logarithm, so that assuming an exponential prior

$$P(\mu) \propto \mathrm{e}^{-\lambda\phi(\mu)} \quad (\lambda > 0),$$

amounts to replacing (4.8) with the penalized problem

$$\underset{\mu}{\text{minimize}}\ f(\mu) \text{ subject to } \mu \geq 0, \qquad f(\mu) := \tfrac{1}{2}\left\|P\mu - y\right\|^2 + \lambda\phi(\mu). \tag{4.10}$$

Hence, under our assumptions and approximations, the reconstruction of an image can be cast as the solution of the bound-constrained regularized linear least-squares problem (4.10).

We favor penalty functions that preserve convexity of the objective of (4.10) and that *smoothen* the attenuation coefficients, i.e. that penalize strong local variations. To this end, Goussard *et al.* (2013) employ $\mathscr{L}_2$ or $\mathscr{L}_2\mathscr{L}_1$ penalty functions, either directly on $\mu$ or on the difference between neighboring voxels. If the $\mathscr{L}_2$ norm is used, we refer to the first case as a *penalty on the object*, and to the second as a *penalty on the gradient of the object*.

The $\mathscr{L}_2$-penalty can be written

$$\phi_{\mathscr{L}_2}(\mu) = \tfrac{1}{2} \sum_{k=1}^{n_{\text{dim}}} \mu^{\mathsf{T}} {D^{(k)}}^{\mathsf{T}} \Gamma^{(k)} D^{(k)} \mu,$$

where $\Gamma^{(k)}$, $k = 1, \ldots, n_{\text{dim}}$, are diagonal weight matrices corresponding to volume elements for each voxel, i.e. they take into account the variable size of each voxel, and $D^{(k)}$, $k = 1, \ldots, n_{\text{dim}}$, are the identity if the penalty applies to the object, or first-derivative matrices if the penalty applies to the gradient of the object.

Note that the superscript $(k)$ indicates the $k$-th matrix and not powers. The $\mathscr{L}_2\mathscr{L}_1$-penalty can be written

$$\phi_{\mathscr{L}_2\mathscr{L}_1}(\mu) = \sum_{k=1}^{n_{\text{dim}}} e^{\mathsf{T}} \Gamma^{(k)} \left( \delta^2 e + (D^{(k)}\mu)^2 \right)^{1/2}$$

where $\delta$ is a nonzero real parameter, $e$ is the vector of ones, and the square and square root are applied componentwise to vectors.

### 4.2.4 Scaled Problem in Cylindrical Coordinates

As indicated in section 4.1, Thibaudeau *et al.* (2013) and Goussard *et al.* (2013) achieved large gains in memory requirements and reconstruction time by discretizing $\mu$ in cylindrical coordinates. Cylindrical coordinates allow us to benefit from geometric redundancies in the data acquisition process, which translate into a *block-circulant* structure for the projection matrix. Hence, only a single row of blocks of $P$ needs to be stored, which greatly reduces the memory footprint (generally by a factor of several hundreds with respect to storage of the full projection matrix in Cartesian coordinates).

Unfortunately, Goussard *et al.* (2013) and Golkar (2013) show that state-of-the-art methods, such as L-BFGS-B (Byrd *et al.*, 1995; Zhu *et al.*, 1997), converge slowly near the origin when applied to (4.10) due to the poor conditioning of $P$. Observe that $P^{\mathsf{T}}P$ is block circulant and that $\phi$ can be chosen so the Hessian of $f$ in (4.10) remains block circulant. Using the

fact that block-circulant matrices may be block-diagonalized by Fourier transforms—see, e.g., (Petersen et Pedersen, 2007), there exists a Hermitian, block-diagonal and positive-definite matrix $\Pi$ such that

$$\nabla^2 f(\mu) = P^\mathsf{T} P + \lambda \nabla^2 \phi(\mu) = \tfrac{1}{n} \mathscr{F}_n^\star \, \Pi \, \mathscr{F}_n,$$

where $\mathscr{F}_n$ is a discrete Fourier transform (DFT) and $n = n_{\text{vox}}$. Golkar (2013) proposes to seek a diagonal and positive-definite approximation $\Delta \approx \Pi$ and to perform the diagonal scaling in Fourier space

$$C = \tfrac{1}{n} \mathscr{F}_n^\star \, \Delta^{-1/2} \, \mathscr{F}_n, \tag{4.11}$$

in hopes to improve the conditioning of (4.10). The simple choice $\Delta = \operatorname{diag}(\Pi)$ has proved effective in practice and is what we use in our implementation.

The change of variables $\mu = Cx$ leads to the *scaled problem*

$$\underset{x}{\text{minimize}} \; \tfrac{1}{2} \left\| PCx - y \right\|^2 + \lambda \phi(Cx) \;\; \text{subject to } Cx \geq 0. \tag{4.12}$$

The disadvantage of (4.12) is that the scaling converts simple bounds into linear inequality constraints.

To illustrate the importance of rescaling the problem (4.10), we report approximate condition numbers $\kappa$ in cylindrical and Cartesian coordinates on a 2D problem of 128×128 pixels in figure 4.1 for various values of the penalty parameter $\lambda$. Our approximations are lower bounds on the actual condition number because computing eigenvalues is time intensive, even for such a small problem. The example of section 4.6 uses a finer resolution, which results in smaller voxels near the origin, and is likely to have worse conditioning. 4.1 shows that both regularization and scaling have the potential to improve the condition number of the Hessian radically.

In a practical implementation, $\mathscr{F}_n$ and $\mathscr{F}_n^\star$ can be applied to a vector in $\mathcal{O}(n \log n)$ time by way of the FFT (Cooley et Tukey, 1965), even though they would materialize as dense matrices. Thus, $C$ can also be applied to a vector in $\mathcal{O}(n \log n)$ time. The presence of such *fast operators* are one of the reasons why it is necessary to devise factorization-free methods for (4.12). Leveraging the nature of $C$ also allows us to design efficient projection operations, which is why we focus our attention on projection-based active-set methods.

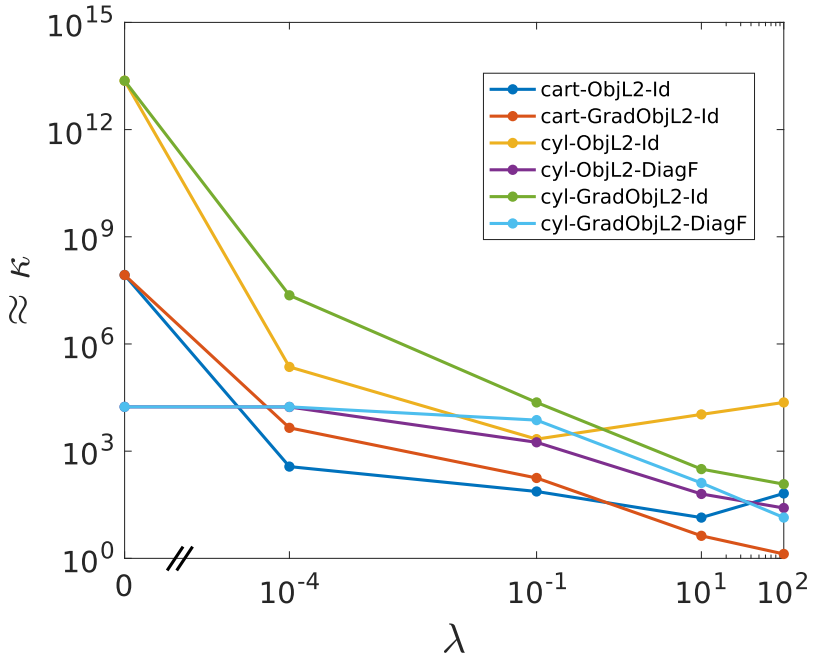



Figure 4.1 Condition number estimate ($\kappa$) of the Hessian as a function of $\lambda$. We compare the Hessian in Cartesian ("cart") and cylindrical ("cyl") coordinates for $\mathscr{L}_2$ penalty functions on the object ("ObjL2") and on the gradient of the object ("GradObjL2") using the scaling matrix ("DiagF") or not ("Id"). Note that the scaling matrix (4.11) only applies in cylindrical coordinates.

### 4.3 Primal Active-Set Methods

Consider a generic optimization problem with linear inequality constraints

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \quad \text{subject to } a_i^{\mathsf{T}}x \geq b_i,\ i = 1,\ \ldots,\ m, \tag{4.13}$$

and assume that $x^*$ is a local solution. Primal active-set methods are designed from the principle that were the optimal active-set at $x^*$ $\mathcal{A}(x^*)$ known ahead of time, it would suffice to solve the equality-constrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \quad \text{subject to } a_i^{\mathsf{T}}x = b_i,\ i \in \mathcal{A}(x^*).$$

The $k$-th iteration of an active-set method consists in approximately solving the equality-constrained subproblem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ f(x) \quad \text{subject to } a_i^{\mathsf{T}}x = b_i,\ i \in \mathcal{A}(x_k), \tag{4.14}$$

where $\mathcal{A}(x_k) \approx \mathcal{A}(x^*)$. The estimate $\mathcal{A}(x_k)$ is then updated based on local information at the current iterate, including the inactive constraints that are violated, the gradient of $f$, and possibly Lagrange multiplier estimates.

When the constraints of (4.13) are simple bounds, those of (4.14) merely fix a subset of variables. For more general linear inequalities, (4.14) is a problem with linear equality constraints. The convergence of active-set methods relies on the fact that there are only finitely many possible active-set estimates and that, once the correct active-set has been identified, so has a local solution to (4.13). For more information on active-set methods, we refer the interested reader to, e.g., (Luenberger et Ye, 2008, Chapter 12).

The active-set methods that we consider below belong to the family of *projected direction* methods. In such methods, certain projection operations are repeatedly applied along the iterations and it is crucial that they be performed efficiently. Recall that if $\mathcal{V} \subseteq \mathbb{R}^n$ is a closed convex set, and if $\bar{x} \in \mathbb{R}^n$, the projection $\mathcal{P}_{\mathcal{V}}[\bar{x}]$ of $\bar{x}$ into $\mathcal{V}$ is well defined, unique and solves

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\left\|x - \bar{x}\right\|^2 \quad \text{subject to } x \in \mathcal{V}. \tag{4.15}$$

Our algorithms require that we design three different projection operations. The first one consists of projecting a vector into the feasible set of (4.13), whereas the second one consists of projecting a vector into a face of the feasible set of (4.13), i.e. into the feasible set of (4.14). The last one can be interpreted as a mixture of the previous projections and consists

of projecting into a face of the feasible set while maintaining the feasibility of the *inactive* constraints. We now describe how we perform each type of projection in the context of the problem (4.12).

### 4.3.1 Projection into the Polyhedral Feasible Set

We first describe how a projection into the feasible set of (4.12),

$$\mathcal{F} = \{x \mid Cx \geq 0\}, \tag{4.16}$$

may be computed efficiently. The projection $\mathcal{P}_{\mathcal{F}}[\bar{x}]$, with $\bar{x} \in \mathbb{R}^n$, solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\left\|x - \bar{x}\right\|^2 \quad \text{subject to } Cx \geq 0, \tag{4.17}$$

where $C$ is given by (4.11). Unfortunately, (4.17) appears nearly as difficult as (4.12). However, as a convex problem, (4.17) has a Lagrange dual—see, e.g., (Boyd et Vandenberghe, 2010)—whose objective is

$$g(z) = \inf_x \ \tfrac{1}{2}\left\|x - \bar{x}\right\|^2 - z^{\mathsf{T}} C x,$$

where $z \geq 0$ are Lagrange multipliers associated to the constraints of (4.17) and the argument of the infimum is the Lagrangian of (4.17). The infimum is attained for $x = \bar{x} + C^{\mathsf{T}} z$, which, when injected into the definition of $g$, yields

$$g(z) = -\tfrac{1}{2}\|C^{\mathsf{T}} z + \bar{x}\|^2 + \tfrac{1}{2}\|\bar{x}\|^2.$$

If we neglect constant terms in the objective function, the Lagrange dual of (4.17), which consists in maximizing $g(z)$ subject to $z \geq 0$, may be written

$$\underset{z \in \mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\|C^{\mathsf{T}} z + \bar{x}\|^2 \quad \text{subject to } z \geq 0. \tag{4.18}$$

The dual problem (4.18) is a bound-constrained linear least-squares problem with a fast operator. If we identify a solution $z^\star$ of (4.18), we may recover a solution of (4.17), i.e. $\mathcal{P}_{\mathcal{F}}[\bar{x}]$, as $x^\star = \bar{x} + C^{\mathsf{T}} z^\star$. We employ primal active-set methods, such as those presented in section 4.5, to solve (4.18) and note that the projection of any $\bar{z} \in \mathbb{R}^n$ into the nonnegative orthant can be computed easily as the componentwise $\max(\bar{z}, 0)$.

### 4.3.2 Projection into the Active Face of the Polyhedral Feasible Set

We now describe how a projection into the *active* face of (4.16),

$$\mathcal{A} = \{x \mid BCx = 0\}, \tag{4.19}$$

may be computed efficiently, where the *restriction operator* $B$ contains the rows of the identity corresponding to the active-set $\{i \mid c_i^T x = 0\}$. The projection $\mathcal{P}_{\mathcal{A}}[\bar{x}]$ solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}\left\|x - \bar{x}\right\|^2 \quad \text{subject to } \ BCx = 0. \tag{4.20}$$

The optimality conditions of (4.20) may be written as the symmetric indefinite system

$$\begin{bmatrix} I & C^T B^T \\ BC & \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \bar{x} \\ 0 \end{bmatrix}, \tag{4.21}$$

where $y$ is the vector of Lagrange multipliers associated to the constraints of (4.20), or, equivalently, as the smaller symmetric positive-definite system

$$BCC^T B^T y = BC\bar{x}. \tag{4.22}$$

Alternatively, we also recognize (4.21) and (4.22) as the optimality conditions of the Lagrange dual of (4.20), which is the unconstrained linear least-squares problem

$$\underset{y}{\text{minimize}} \ \tfrac{1}{2}\|C^T B^T y - \bar{x}\|^2, \tag{4.23}$$

and from which we recover $x = \bar{x} - C^T B^T y$.

In our implementation, we consider several possible ways of solving (4.20). The first is to solve (4.21) using MINRES (Paige et Saunders, 1975), an iterative method for symmetric, not necessarily definite, linear systems that ensures that the system residual decreases monotonically. The second is to solve (4.22) with PCG (Hestenes et Stiefel, 1952) or MINRES, which is reasonable in this case because the system is defined by a fast operator that we expect to have a moderate condition number. The third is to solve (4.23) using an iterative method for linear least-squares problems such as LSQR (Paige et Saunders, 1982) or LSMR (Fong et Saunders, 2011). By design, LSQR and LSMR applied to (4.23) are equivalent to PCG and MINRES applied to (4.22), respectively, in exact arithmetic, but should be more accurate and less sensitive to ill-conditioning in finite precision. Early experiments suggest that the first approach does not show any advantage compared to the other two and typically

requires more storage. Thus we do not consider it further in the sequel.

### 4.3.3 Projection into the "Mixed" Set

The last type of projection we encounter consists of projecting into the active face (4.19), while ensuring that the remaining *inactive* constraints remain satisfied. This "mixed" set

$$\mathcal{M} = \left\{ x \;\middle|\; BCx = 0,\ ACx \geq 0 \right\} \tag{4.24}$$

is a mixture of the two previous sets, where $B$ is the restriction matrix defined in section 4.3.2 and $A$ is the restriction matrix that contains the rows of the identity corresponding to the inactive constraints $\{i \mid c_i^T x > 0\}$. The projection $\mathcal{P}_{\mathcal{M}}[\bar{x}]$ solves

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\left\|x - \bar{x}\right\|^2 \quad \text{subject to}\ BCx = 0,\ ACx \geq 0, \tag{4.25}$$

which is a special case of (4.17) where part of the Lagrange multipliers correspond to equality constraints and are free. The dual of (4.25) is

$$\underset{z \in \mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\|C^T z + \bar{x}\|^2 \quad \text{subject to}\ Az \geq 0. \tag{4.26}$$

Thus, solving (4.26) is at most as difficult as solving (4.18) and can be done using the same techniques.

## 4.4 Solving the Reconstruction Problem

In this section, we present the methods that we consider to solve problem (4.12). We first consider a spectral projected gradient (SPG) method (Birgin *et al.*, 2001). First-order methods are attractive due to their low computational cost per iteration, i.e. they don't require Hessian products and only require projections on the feasible set (4.16). Their main drawback is slow convergence near the minimum, which often materializes as short "zig-zagging" steps. Even though those methods seem beneficial from a computational point of view, they might require considerably more work than second-order methods to reach a solution. Thus we also develop a novel adaptation of the Newton trust-region solver TRON (Lin et Moré, 1999) to the case of linear inequality constraints defined by a fast operator. Our adaptation of TRON requires the three different types of projection described in sections 4.3.1, 4.3.2 and 4.3.3 respectively.

### 4.4.1 Non-Monotone Spectral Projected-Gradient Method

The non-monotone spectral projected gradient (SPG) algorithm described in algorithm 5 is based on the refinements to the original algorithm of Barzilai et Borwein (1988) proposed by Birgin *et al.* (2001) and Birgin et Martínez (2002). Projected-gradient methods are suited to problems with a closed convex feasible set such that projections into this set are inexpensive. In our case, we consider (4.12) and compute projections into the feasible set $\mathcal{F}$ using the procedure outlined in section 4.3.1.

**Algorithm 5** Non-monotone spectral projected gradient

1: Initialize $x_0 = \mathcal{P}_{\mathcal{F}}[x_0]$, $k = 0$, $\alpha_0 = 1$ and $g_0 = \nabla f(x_0)$
2: **while** stopping criteria not met **do**
3: $d_k = \mathcal{P}_{\mathcal{F}}[x_k - \alpha_k g_k] - x_k$
4: compute $\lambda_k$ using 8
5: update $x_{k+1} = x_k + \lambda_k d_k$ and $g_{k+1} = \nabla f(x_{k+1})$
6: set $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$
7: update $\alpha_k$ using either algorithm 9, 10 or 11
8: $k = k + 1$
9: **end while**

Some of the details of algorithm 5 are stated in (4.8.1). 5 uses Barzilai-Borwein step lengths and a non-monotone Armijo linesearch in conjunction with the projected gradient step.

Barzilai et Borwein (1988) proposed two step lengths. Birgin *et al.* (2001) recommend using only one of the two step lengths, but Dai et Fletcher (2005), Frassoldati *et al.* (2008), Bonettini *et al.* (2009) and di Serafino *et al.* (2017) employ alternating step length strategies in order to further improve convergence. We use the *adaptive* step length schemes that provide rules for choosing between the two step lengths with dynamic thresholds, such as algorithms 10 and 11.

### 4.4.2 TRON for Linear Inequalities

Lin et Moré (1999) argue that the convergence theory of TRON for bound constraints continues to apply to the case of a linear inequality constraints. In this section, we describe our adaptation of TRON to linear inequalities and emphasize the most costly subproblems. We refer the reader to (Lin et Moré, 1999) for a complete description of the algorithm. Our contention is that when the linear inequality constraints are defined by a fast operator, an efficient implementation remains possible.

TRON is a trust-region active-set method for problems of the form (4.13). At each iteration,

we construct a quadratic approximation of the objective,

$$\psi(w) = g_k^T w + \tfrac{1}{2} w^T H_k w, \tag{4.27}$$

where $g_k = \nabla f(x_k)$ and $H_k = \nabla^2 f(x_k)$. A central subproblem consists in minimizing $\psi(w)$ on the *active face* of the feasible set, while respecting a trust-region bound. Lin et Moré (1999) establish convergence by requiring that the decrease in the quadratic approximation achieved by a trial step be at least a fraction of the decrease achieved by an approximate *Cauchy step*.

An approximate Cauchy step, denoted $s^C$, is an approximate minimizer of $\psi$ along the projected-gradient direction, i.e.,

$$s^C = \mathcal{P}_{\mathcal{F}}[x_k - \alpha g_k] - x_k, \tag{4.28}$$

where the feasible set $\mathcal{F}$ is defined in (4.16), such that $\alpha$ achieves sufficient decrease, and such that the trust-region bound is satisfied. We employ the same strategy as Lin et Moré (1999) to obtain $s^C$, except that we limit the number of extrapolation steps, which play no role in guaranteeing convergence, but help take larger steps. Both interpolation and extrapolation steps require one projection and one Hessian product, both of which are costly in our case.

Before tackling the trust-region subproblem, which is the core of the algorithm, we define the *active-set*, representing the constraints that are at their bound at $x_k$, and the *breakpoints*, which are the step lengths along a direction $w$ from $x_k$ such that previously *inactive* constraints become *active*.

### Computing the Active-Set and Breakpoints for Linear Inequalities

Given the feasible set (4.16) and $x_k \in \mathcal{F}$, the active-set at $x_k$ is

$$\mathcal{A}(x_k) = \{i \mid c_i^T x_k = 0\}, \tag{4.29}$$

where $c_i^T$ is the $i$-th row of $C$.

To compute the breakpoints, we first find the set of indices of inactive constraints that could become violated—or active—if we take a step $w$ from $x_k$:

$$\mathscr{L} = \{i \mid c_i^T x_k > 0 \text{ and } c_i^T w < 0\}. \tag{4.30}$$

We then compute the positive step lengths $\alpha_{\mathscr{L}}$ along $w$ such that at least one additional

constraint becomes active for the indices (4.30):

$$\alpha_{\mathscr{L}i} = -\frac{c_i^T x_k}{c_i^T w} \quad (i \in \mathscr{L}), \tag{4.31}$$

where $\alpha_{\mathscr{L}i} > 0$ denotes the $i$-th component of $\alpha_{\mathscr{L}}$. Because our interest in the breakpoints is to put boundaries on interpolation and extrapolation steps, we pick the minimal and maximal values of (4.31).

An additional difficulty is that, in general, we cannot project into $\mathcal{F}$ *exactly*; we must solve (4.18) approximately. We employ a relative measure $\epsilon$ defined by a *projection tolerance*, noted $\epsilon_{\text{proj}}$, the norm of $C$ and the norm of $x_k$:

$$\epsilon = \epsilon_{\text{proj}} \cdot \left\|C\right\| \cdot \left\|x_k\right\|. \tag{4.32}$$

In other words, we use a *relaxed* definition of (4.29) and (4.30). Given (4.11), we have $\|C\| = \left\|\Delta\right\|^{-1/2}$ (we do not include the factor $1/n$ in the constraint definition), which is readily available. Nearly-active constraints at $x_k$ are those for which $|c_i^T x_k| \leq \epsilon$. We stress that the correct identification of the active-set proves to be critical in practice and the implementation is highly sensitive to the value of $\epsilon_{\text{proj}}$.

### Solving the Trust-Region Problem

Lin et Moré (1999) find an approximate minimizer of (4.27) on the active face and inside the trust region by constructing a sequence of *minor iterates*. Generating the latter not only allows to update the active-set at each iteration, but also to take *projected steps* in order to *add* more constraints to the active-set. The purpose of this strategy is to quickly determine whether the minimizer of $\psi$ lies in the current active face. Minor iterates are defined so as to ensure both the convergence of the method and feasibility with respect to the feasible set and the trust region. More precisely, a minor iterate $x_{k,j}$ is defined as

$$x_{k,j+1} = \mathcal{P}_{\mathcal{M}}[x_{k,j} + \alpha w^\star] \tag{4.33}$$

where $\mathcal{M}$ is the set (4.24), $x_{k,1} := x_k + s^C$ and $w^\star$ is chosen such that

$$\left\|w^\star + x_{k,j} - x_k\right\| \leq \Delta.$$

We refer the reader to (Lin et Moré, 1999) for more details. In order to compute (4.33), we use the procedure outlined in section 4.3.3. Finally, $x_{k,j+1}$ must also satisfy a sufficient

decrease condition and a suitable $\alpha$ is obtained by way of a linesearch. We now describe how $w^\star$ is obtained as an approximate minimizer of the trust-region subproblem.

For bound-constrained problems, reducing a problem to the active face simply implies *fixing* a subset of variables. Hence, $w^\star$ can be obtained by minimizing $\psi(w)$ subject to the trust region on the free variables. Lin et Moré (1999) employ the truncated conjugate gradient method (TRCG) proposed by Steihaug (1983). In our case, such simplification is not possible. We obtain $w^\star$ as an approximate solution of the quadratic optimization problem with linear equality and trust-region constraints

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \psi(w+s) \\ \text{subject to} \quad & BCw = 0 \\ & \left\|w+s\right\| \leq \Delta, \end{aligned} \tag{4.34}$$

where $s = x_{k,j} - x_k$ represents the current step from $x_k$, and $B$ is the restriction matrix composed of the rows of the identity corresponding to indices in $\mathcal{A}(x_{k,j})$.

We solve (4.34) using the projected truncated conjugate-gradient method (Gould *et al.*, 2001, 2013), which essentially amounts to applying TRCG and projecting each conjugate-gradient search direction into $\mathcal{A}$.

Once $w^\star$ is obtained, the next minor iterate is computed using (4.33). Lin et Moré (1999) propose to terminate the procedure if either $\left\|x_{k,j} - x_k\right\| > \Delta$ or if the decrease condition

$$\|Z^\mathsf{T} \nabla\psi(s)\| \leq \epsilon_{\text{grad}} \|Z^\mathsf{T} g_k\|$$

is satisfied, where $\epsilon_{\text{grad}} > 0$ is a tolerance, typically 1e$-$2, and the columns of $Z$ form an orthonormal basis for the nullspace of $BC$—note that the implementation never needs to compute $Z$ explicitly. The former simply implies that we have crossed the trust-region boundary, and the latter that $s$ satisfies a sufficient decrease condition for the inactive variables.

### TRON Algorithm

In summary, we extend the algorithm of Lin et Moré (1999) to the case of linear inequality constraints. The rules governing the update of the trust-region radius and to determine whether a step is valid are the same as those of Lin et Moré (1999). If a step is rejected, we added the option of an Armijo backtracking linesearch. algorithm 6 summarizes the projected Newton method.

**Algorithm 6** TRON for linear inequalities

Initialize $x_0 = \mathcal{P}[x_0]$, $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$
**while** stopping criteria on (4.12) not satisfied **do**
  Compute $\psi(w)$ (4.27)
  Compute the Cauchy step (4.28)
  **while** $\left\|x_{k,j} - x_k\right\| \leq \Delta$ and $\|Z^\mathsf{T} \nabla\psi(s)\| > \epsilon_{\text{grad}} \|Z^\mathsf{T} g_k\|$ **do**
    Update the active-set (4.29)
    Solve (4.34) using the projected TRCG method
    Compute the next minor iterate (4.33)
  **end while**
  Update the trust-region radius and either accept or reject the step
**end while**

## 4.5 Solving the Projection subproblem

Our current implementation requires accurate solutions of (4.18) and (4.26), which we anticipate first-order methods may not be able to achieve. In order to validate this claim, we compare SPG, presented in section 4.4.1, with second-order methods for the solution of (4.18) and (4.26). The other solvers that we consider are the *two-metric projection* (TMP) algorithm of Gafni et Bertsekas (1984), the original trust-region Newton solver (TRON) of Lin et Moré (1999), and the L-BFGS-B solver of Zhu *et al.* (1997). Since TRON and SPG were detailed previously, we now detail our implementation of TMP. To a reasonable extent, L-BFGS-B may be understood as a special case of TMP.

### 4.5.1 Two-Metric Projection Algorithm

The two-metric projection algorithm (TMP) of Gafni et Bertsekas (1984) is similar to the projected Newton method of Bertsekas (1982). Broadly speaking, two-metric projection methods are active-set methods that use second-order information to *rescale* the steepest descent direction. However, when the feasible set is defined by simple bounds $l \leq x \leq u$, simplifications occur and stronger convergence results can be obtained using the *binding set* instead of the *active-set*. The binding set comprises the indices of the variables that are at their bound and are likely to remain there were a gradient-descent step taken:

$$\mathcal{B}(x_k) = \{i \mid (x_{ki} = l_i \text{ and } g_{ki} > 0) \text{ or } (x_{ki} = u_i \text{ and } g_{ki} < 0)\}, \tag{4.35}$$

where $x_{ki}$ and $g_{ki}$ are the $i$-th components of $x_k$ and $g_k$. A *rescaled* gradient step $d_k$ is then computed for the *free* variables, i.e., those not in $\mathcal{B}(x_k)$, as the solution of

$$\underset{d_k}{\text{minimize}} \ \tfrac{1}{2} d_k^T Q_k d_k + g_k^T d_k \quad \text{subject to } \ d_{ki} = 0, \ i \in \ \mathcal{B}(x_k), \tag{4.36}$$

where $Q_k$ is a symmetric approximation of the Hessian of the objective—possibly a quasi-Newton approximation—such that $Q_k$ is positive definite when restricted to the subspace of free variables. Such methods regained interest recently in the context of large-scale bound-constrained problems, including those arising from machine learning (Schmidt *et al.*, 2011). In our experience, whereas quasi-Newton approximations of the Hessian $H_k$ may yield faster computations, attaining *tighter* optimality tolerances often proves to be impossible. Thus, we allow $Q_k = H_k$—the Hessian of the objective function—and provide Krylov methods to solve (4.36). The optimality conditions of (4.36) may be written as the linear system

$$(BH_kB^T)Bd_k = -Bg_k, \tag{4.37}$$

where $B$ is the restriction matrix formed with the rows of the identity corresponding to indices $i \notin \mathcal{B}(x_k)$ and the remaining components of $d_k$ are set to zero. As in section 4.3.2, candidate Krylov methods for (4.37) include PCG (Hestenes et Stiefel, 1952) and MINRES (Paige et Saunders, 1975), as well as their counterparts LSQR (Paige et Saunders, 1982) and LSMR (Fong et Saunders, 2011). However, given that LSQR and LSMR handle (4.37) as a linear least-squares problem, they cannot benefit from the fact that the Hessian $H_k$ of (4.18) or (4.26) is

$$H_k = CC^T = \left(\tfrac{1}{n}\, \mathscr{F}_n^\star\, \Delta^{-1/2}\, \mathscr{F}_n\right) \left(\tfrac{1}{n}\, \mathscr{F}_n^\star\, \Delta^{-1/2}\, \mathscr{F}_n\right)^T = \tfrac{1}{n}\, \mathscr{F}_n^\star\, \Delta^{-1}\, \mathscr{F}_n,$$

thus saving two FFT operations. Moreover, whereas (4.22) had to be solved once to project into the active face, (4.37) must be solved at each iteration of TMP. For that reason, we recommend PCG and MINRES.

7 summarizes the two-metric projection algorithm that we consider, where $\mathcal{P}_+$ is the projection into the nonnegative orthant.

### 4.6 Numerical Results

In this section, we compare SPG and TRON in order to identify which performs best on (4.12). Given that projections play a crucial role in our implementations, we are also interested in determining which combination of solvers leads to both more accurate projections and

**Algorithm 7** Two-metric projection algorithm for bounded problem

1: Initialize $x_0 = \mathcal{P}_+[x_0]$, $k = 0$ and $g_0 = \nabla f(x_0)$
2: **while** stopping criteria not met **do**
3: compute the binding set (4.35)
4: compute the descent direction (4.36)
5: compute the new iterate $x_{k+1} = \mathcal{P}_+[x_k + \alpha_k d_k]$ where $\alpha_k > 0$ is obtained via a projected Armijo linesearch.
6: **end while**

shorter run times. To that effect, we compare various combinations of reconstruction solvers and projection solvers for the solution of (4.12). Synthetic projection data, i.e. intensity measurements, as well a realistic human phantom, are provided by the software XCAT, developed by Segars *et al.* (2008). The geometric parameters and X-ray source characteristics were chosen to emulate a standard clinical scanner, with a measurement quality comparable to that of standard acquisition protocols. Using different techniques for data generation and reconstruction allows us to avoid the *inverse crime* Wirgin (2004) to a large extent.

We consider a 2D example consisting of a slice of abdomen, shown in 4.2, and aim to reconstruct it from the corresponding intensity measurements. Our example contains 512×512 square pixels of length 0.7 mm each, and the sinogram was obtained from 672 detectors and $1,160$ scan angles. In Cartesian coordinates, $n_{\text{meas}} = 672 \cdot 1,160 = 779,520$ and $n_{\text{vox}} = 512^2 = 262,144$. In cylindrical coordinates, recall that we assume that the number of angular elements must be an integer multiple of the scan angles. Say we consider $1,160$ angular elements, the number of radial elements is $512^2/1,160 \approx 225.9$, which we round up to 226. The effective number of voxels is then $n_{\text{vox}} = 226 \times 1160 = 262,160$. Thus, we have slightly more voxels in cylindrical coordinates due to the size difference between polar and square pixels.

We study the impact of the penalty function by comparing $\mathscr{L}_2$ penalty functions on the gradient and on the object respectively. Since the $P$ operators and the first-differences matrices used with the penalty on the gradient are fundamentally different in cartesian and cylindrical coordinates and unnormalized, different penalty parameters have to be used in order to produce images of similar quality. Thus, we proceed empirically by comparing various reconstruction results to determine appropriate values of $\lambda$. This lead to $\lambda = 5\text{e}{-2}$ in cylindrical coordinates and $\lambda = 10$ and $\lambda = 25$ in cartesian coordinates using a penalty function on the gradient of the object and the object respectively. For each solver, we use the norm of the projected gradient as an optimality measure. For (4.10) and (4.12), we use

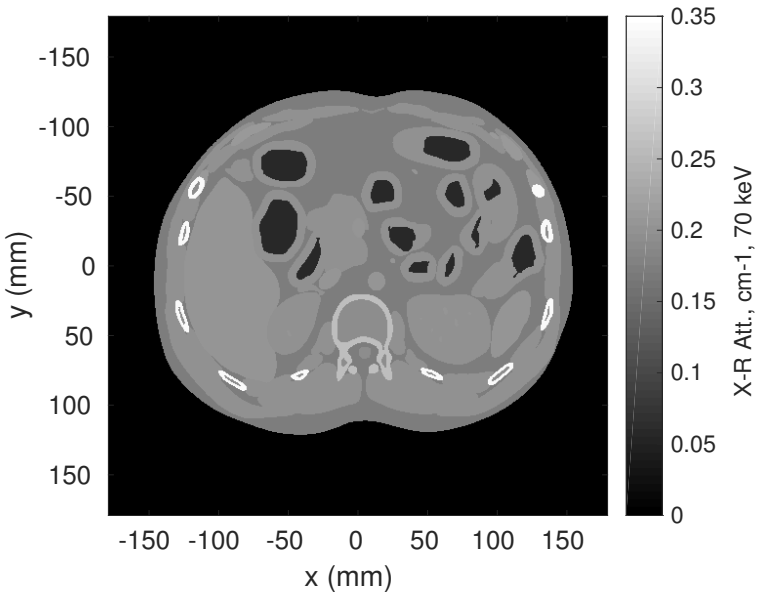



Figure 4.2 Original phantom: slice of abdomen, 512×512 pixels of size 0.7 mm × 0.7 mm. Sinogram obtained from 672 detectors and 1,160 projection angles.

the stopping condition

$$\|x_k - \mathcal{P}[x_k - g_k]\| \leq \epsilon_a + \epsilon_r \|x_0 - \mathcal{P}[x_0 - g_0]\|,$$

with $\epsilon_a = \epsilon_r = 1\text{e}{-8}$ and where $\mathcal{P}$ is either $\mathcal{P}_+$ or $\mathcal{P}_{\mathcal{F}}$. We set $\epsilon_{\text{proj}} = 1\text{e}{-11}$ for the projection operations (4.18), (4.20) and (4.26). We set a tolerance of 1e−15 on the progress to avoid stagnation in case the norm of the search direction or the improvement in the objective becomes too small. Finally, we limit the total run time to 15 minutes and allow a maximum of 5 minutes for any one projection operation.

To further validate our results, we compare our algorithms with L-BFGS-B applied to (4.10) in *Cartesian coordinates*, which is our benchmark and offers the best results according to Hamelin (2009). Our goal is to obtain similar performance as L-BFGS-B, while radically reducing the memory footprint. In this specific example, the projection matrices in Cartesian coordinates and cylindrical coordinates require about 1.0Gb and 4.4Mb, respectively, i.e., cylindrical coordinates allow savings of a factor of about 233. However, keep in mind that (4.10) in Cartesian coordinates and (4.12) in cylindrical coordinates are different problems altogether. All solvers are implemented in MATLAB except L-BFGS-B, which is in C++.

The tables of results use the following nomenclature. The columns $\mathcal{P}_{\mathcal{A}}$ and $\mathcal{P}_{\mathcal{F}}$ indicate the choice of method to compute the corresponding projection. The column labeled KKT gives

the final optimality residual $\|x_k - \mathcal{P}[x_k - g_k]\|$, "time" gives the run time in seconds, $f(x^*)$ is the final objective value, $\|\mu < 0\|$ is the final infeasibility, "#$P$" and "#$C$" are the number of operator-vector products with $P$ and $C$, respectively, and their adjoints, "#iter" is the number of iterations, "#$\mathcal{P}_\mathcal{F}$" and "#$\mathcal{P}_\mathcal{A}$" are the number of projections into the feasible set and into an active face. The numbers of projections $\mathcal{P}_\mathcal{F}$ and $\mathcal{P}_\mathcal{M}$ are grouped together given the similarity of the projection problems. The failure codes reported are as follows: "cpu" indicates that the maximum run time was exceeded, "prog" indicates that the improvement in the objective function was too small and "xfail" indicates an unknown failure.

(4.1), (4.2), and (4.3) report the results of L-BFGS-B on (4.10) and TRON and SPG on (4.12) using a penalty on the gradient of the object.

Table 4.1 – L-BFGS-B applied to (4.10) using a $\mathscr{L}_2$ penalty function on the gradient of the object with $\lambda = 15$.

| Proj. Solver | KKT | time | $f(x^\star)$ | $\|\mu < 0\|$ | #P | #C | #iter | #$\mathcal{P}_\mathcal{F}$ | #$\mathcal{P}_\mathcal{A}$ | fail |
|---|---|---|---|---|---|---|---|---|---|---|
| N/A | 3.4e−6 | 2.0e2 | 6.8e3 | 0 | 136 | N/A | 61 | N/A | N/A | |

Table 4.2 – TRON reconstruction results on (4.12) using a $\mathscr{L}_2$ penalty function on the gradient of the object with $\lambda = 0.1$.

| $\mathcal{P}_\mathcal{A}$ | $\mathcal{P}_\mathcal{F}$ | KKT | time | $f(x^\star)$ | $\|\mu < 0\|$ | #$P$ | #$C$ | #iter | #$\mathcal{P}_\mathcal{F}$ | #$\mathcal{P}_\mathcal{A}$ | fail |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LSQR | TMP-PCG | 3.6e−5 | 1.5e2 | 6.9e3 | 2.2e−14 | 240 | 5914 | 9 | 46 | 86 | |
| | TMP-MINRES | 3.6e−5 | 1.4e2 | 6.9e3 | 2.1e−14 | 240 | 5840 | 9 | 46 | 86 | |
| | SPG | 2.0e−5 | 1.4e2 | 6.9e3 | 1.1e−8 | 240 | 5527 | 9 | 46 | 86 | |
| | TRON | 3.6e−5 | 4.6e2 | 6.9e3 | 3.2e−14 | 240 | 33337 | 9 | 46 | 86 | |
| | L-BFGS-B | 3.5e−5 | 1.4e2 | 6.9e3 | 2.4e−7 | 246 | 4619 | 9 | 47 | 87 | |
| LSMR | TMP-PCG | 3.6e−5 | 1.5e2 | 6.9e3 | 2.1e−14 | 240 | 5908 | 9 | 46 | 86 | |
| | TMP-MINRES | 3.6e−5 | 1.5e2 | 6.9e3 | 2.1e−14 | 240 | 5834 | 9 | 46 | 86 | |
| | SPG | 2.4e−4 | 1.4e2 | 6.9e3 | 1.7e−9 | 250 | 5405 | 9 | 46 | 86 | prog. |
| | TRON | 3.6e−5 | 4.6e2 | 6.9e3 | 3.1e−14 | 240 | 29182 | 9 | 46 | 86 | |
| | L-BFGS-B | 3.5e−5 | 1.4e2 | 6.9e3 | 2.4e−7 | 246 | 4645 | 9 | 47 | 87 | |
| MINRES | TMP-PCG | 3.6e−5 | 1.4e2 | 6.9e3 | 3.6e−9 | 240 | 4471 | 9 | 46 | 86 | |
| | TMP-MINRES | 3.6e−5 | 1.4e2 | 6.9e3 | 3.6e−9 | 240 | 4397 | 9 | 46 | 86 | |
| | SPG | 3.0e−5 | 1.3e2 | 6.9e3 | 3.8e−9 | 242 | 3997 | 9 | 45 | 87 | |
| | TRON | 3.6e−5 | 4.5e2 | 6.9e3 | 3.6e−9 | 240 | 27164 | 9 | 46 | 86 | |
| | L-BFGS-B | 3.5e−5 | 1.2e2 | 6.9e3 | 2.4e−7 | 246 | 3109 | 9 | 47 | 87 | |
| PCG | TMP-PCG | 3.6e−5 | 1.5e2 | 6.9e3 | 1.7e−11 | 240 | 4855 | 9 | 46 | 86 | |
| | TMP-MINRES | 3.6e−5 | 1.4e2 | 6.9e3 | 1.7e−11 | 240 | 4781 | 9 | 46 | 86 | |
| | SPG | 2.0e−5 | 1.3e2 | 6.9e3 | 2.8e−8 | 238 | 4240 | 9 | 46 | 84 | |
| | TRON | 3.6e−5 | 4.6e2 | 6.9e3 | 1.7e−11 | 240 | 27524 | 9 | 46 | 86 | |
| | L-BFGS-B | 3.5e−5 | 1.3e2 | 6.9e3 | 2.4e−7 | 246 | 3523 | 9 | 47 | 87 | |

Similarly, (4.4), (4.5) and (4.6) compare L-BFGS-B, TRON and SPG using a penalty on the object.

For both penalty functions, TRON is effective and fast, and is accurate in terms of final infeasibility, especially when combined with either LSQR or LSMR to apply $\mathcal{P}_\mathcal{A}$ and TMP-PCG or TMP-MINRES to apply $\mathcal{P}_\mathcal{F}$. Most combinations perform reasonably, with a few

Table 4.3 – SPG reconstruction results on (4.12) using a $\mathscr{L}_2$ penalty function on the gradient of the object with $\lambda = 0.1$.

| Step | $\mathcal{P}_{\mathcal{F}}$ | KKT | time | $f(x^\star)$ | $\left\|\mu < 0\right\|$ | #P | #C | #iter | $\#\mathcal{P}_{\mathcal{F}}$ | $\#\mathcal{P}_{\mathcal{A}}$ | fail |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BB1 | TMP-PCG | 3.5e−5 | 8.1e2 | 6.9e3 | 2.6e−15 | 610 | 41867 | 176 | 353 | 0 | |
| | TMP-MINRES | 2.8e−4 | 8.6e2 | 6.9e3 | 2.6e−15 | 729 | 41190 | 209 | 419 | 0 | prog. |
| | SPG | 7.9e−4 | 3.7e2 | 6.9e3 | 3.0e−7 | 499 | 18143 | 139 | 279 | 0 | prog. |
| | TRON | 1.4 | 9.0e2 | 6.9e3 | 3.0e−14 | 166 | 54611 | 49 | 99 | 0 | cpu |
| | L-BFGS-B | 6.6e−5 | 6.0e2 | 6.9e3 | 1.8e−7 | 535 | 16948 | 156 | 313 | 0 | |
| ABB | TMP-PCG | 2.5e−4 | 4.8e2 | 6.9e3 | 4.3e−15 | 218 | 26889 | 70 | 141 | 0 | prog. |
| | TMP-MINRES | 2.6e−4 | 4.2e2 | 6.9e3 | 2.5e−15 | 218 | 22370 | 70 | 141 | 0 | prog. |
| | SPG | 4.5e−3 | 1.9e2 | 6.9e3 | 1.0e−6 | 244 | 9889 | 66 | 133 | 0 | prog. |
| | TRON | 2.8e−2 | 9.0e2 | 6.9e3 | 2.3e−14 | 156 | 56879 | 50 | 101 | 0 | cpu |
| | L-BFGS-B | 3.8e−4 | 4.1e2 | 6.9e3 | 1.2e−7 | 262 | 11797 | 77 | 155 | 0 | prog. |
| $\text{ABB}_{\text{min1}}$ | TMP-PCG | 4.2e−4 | 5.0e2 | 6.9e3 | 6.8e−15 | 240 | 27727 | 75 | 151 | 0 | prog. |
| | TMP-MINRES | 4.2e−4 | 4.3e2 | 6.9e3 | 2.6e−15 | 240 | 23065 | 75 | 151 | 0 | prog. |
| | SPG | 1.8e−4 | 2.4e2 | 6.9e3 | 2.4e−7 | 327 | 11937 | 84 | 171 | 0 | prog. |
| | TRON | 7.1e−2 | 9.0e2 | 6.9e3 | 4.0e−7 | 158 | 56808 | 50 | 101 | 0 | cpu |
| | L-BFGS-B | 5.0e−4 | 4.1e2 | 6.9e3 | 1.3e−7 | 259 | 11604 | 74 | 149 | 0 | prog. |
| $\text{ABB}_{\text{SS}}$ | TMP-PCG | 2.2e−4 | 5.2e2 | 6.9e3 | 2.0e−12 | 257 | 29508 | 83 | 167 | 0 | prog. |
| | TMP-MINRES | 5.6e−4 | 4.5e2 | 6.9e3 | 3.3e−15 | 257 | 24537 | 83 | 167 | 0 | prog. |
| | SPG | 9.6e−5 | 2.2e2 | 6.9e3 | 2.7e−9 | 271 | 11693 | 86 | 173 | 0 | |
| | TRON | 7.8e−4 | 6.9e2 | 6.9e3 | 4.8e−15 | 227 | 41694 | 73 | 147 | 0 | prog. |
| | L-BFGS-B | 7.0e−4 | 3.7e2 | 6.9e3 | 8.0e−8 | 249 | 10812 | 79 | 159 | 0 | prog. |

Table 4.4 – L-BFGS-B applied to (4.10) using a $\mathscr{L}_2$ penalty function on the object with $\lambda = 25$.

| Proj. Solver | KKT | time | $f(x^\star)$ | $\left\|\mu < 0\right\|$ | #P | #C | #iter | $\#\mathcal{P}_{\mathcal{F}}$ | $\#\mathcal{P}_{\mathcal{A}}$ | fail |
|---|---|---|---|---|---|---|---|---|---|---|
| N/A | 2.3e−6 | 1.0e2 | 1.2e5 | 0 | 114 | N/A | 49 | N/A | N/A | |

Table 4.5 – TRON reconstruction results on (4.12) using a $\mathscr{L}_2$ penalty function on the object with $\lambda = 0.1$.

| $\mathcal{P}_{\mathcal{A}}$ | $\mathcal{P}_{\mathcal{F}}$ | KKT | time | $f(x^\star)$ | $\left\|\mu < 0\right\|$ | $\#P$ | $\#C$ | #iter | $\#\mathcal{P}_{\mathcal{F}}$ | $\#\mathcal{P}_{\mathcal{A}}$ | fail |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LSQR | TMP-PCG | 1.9e−4 | 1.2e2 | 7.1e4 | 2.8e−14 | 193 | 5740 | 8 | 38 | 63 | |
| | TMP-MINRES | 1.9e−4 | 1.2e2 | 7.1e4 | 2.6e−14 | 193 | 5297 | 8 | 38 | 63 | |
| | SPG | 1.9e−4 | 1.1e2 | 7.1e4 | 2.9e−10 | 193 | 5367 | 8 | 38 | 63 | |
| | TRON | 1.9e−4 | 4.3e2 | 7.1e4 | 6.2e−14 | 193 | 33407 | 8 | 38 | 63 | |
| | L-BFGS-B | 1.9e−4 | 1.2e2 | 7.1e4 | 1.1e−7 | 193 | 4303 | 8 | 38 | 63 | |
| LSMR | TMP-PCG | 1.9e−4 | 1.2e2 | 7.1e4 | 2.5e−14 | 193 | 5738 | 8 | 38 | 63 | |
| | TMP-MINRES | 1.9e−4 | 1.2e2 | 7.1e4 | 2.4e−14 | 193 | 5295 | 8 | 38 | 63 | |
| | SPG | 1.9e−4 | 1.1e2 | 7.1e4 | 3.2e−10 | 193 | 5207 | 8 | 38 | 63 | |
| | TRON | 1.9e−4 | 4.2e2 | 7.1e4 | 6.1e−14 | 193 | 33536 | 8 | 38 | 63 | |
| | L-BFGS-B | 1.9e−4 | 1.3e2 | 7.1e4 | 1.1e−7 | 193 | 4323 | 8 | 38 | 63 | |
| MINRES | TMP-PCG | 1.9e−4 | 1.2e2 | 7.1e4 | 1.2e−15 | 193 | 4885 | 8 | 38 | 63 | |
| | TMP-MINRES | 1.9e−4 | 1.1e2 | 7.1e4 | 1.2e−15 | 193 | 4441 | 8 | 38 | 63 | |
| | SPG | 7.3e−5 | 1.1e2 | 7.1e4 | 4.8e−10 | 222 | 4909 | 9 | 43 | 73 | |
| | TRON | 2.5e−3 | 4.0e2 | 7.1e4 | 4.6e−11 | 160 | 32301 | 7 | 33 | 51 | cpu. |
| | L-BFGS-B | 1.9e−4 | 1.2e2 | 7.1e4 | 1.2e−7 | 193 | 3416 | 8 | 38 | 63 | |
| PCG | TMP-PCG | 1.9e−4 | 1.2e2 | 7.1e4 | 1.0e−11 | 193 | 5139 | 8 | 38 | 63 | |
| | TMP-MINRES | 1.9e−4 | 1.2e2 | 7.1e4 | 1.0e−11 | 193 | 4696 | 8 | 38 | 63 | |
| | SPG | 7.3e−5 | 1.1e2 | 7.1e4 | 4.3e−10 | 222 | 5142 | 9 | 43 | 73 | |
| | TRON | 1.9e−4 | 4.2e2 | 7.1e4 | 1.0e−11 | 193 | 33556 | 8 | 38 | 63 | |
| | L-BFGS-B | 1.9e−4 | 1.2e2 | 7.1e4 | 1.1e−7 | 193 | 3657 | 8 | 38 | 63 | |

Table 4.6 – SPG reconstruction results on (4.12) using a $\mathscr{L}_2$ penalty function on the object with $\lambda = 0.1$.

| Step | $\mathcal{P}_{\mathcal{F}}$ | KKT | time | $f(x^\star)$ | $\|\mu < 0\|$ | $\#P$ | $\#C$ | #iter | $\#\mathcal{P}_{\mathcal{F}}$ | $\#\mathcal{P}_{\mathcal{A}}$ | fail |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BB1 | TMP-PCG | 1.5e−4 | 3.2e2 | 7.1e4 | 2.0e−15 | 406 | 17031 | 118 | 237 | 0 | |
| | TMP-MINRES | 2.8e−4 | 3.2e2 | 7.1e4 | 8.8e−16 | 410 | 16161 | 120 | 241 | 0 | prog. |
| | SPG | 2.1e−4 | 2.6e2 | 7.1e4 | 4.9e−8 | 384 | 15080 | 107 | 215 | 0 | prog. |
| | TRON | 3.3e−5 | 6.1e2 | 7.1e4 | 1.1e−14 | 397 | 43760 | 116 | 233 | 0 | |
| | L-BFGS-B | 1.3e−4 | 3.0e2 | 7.1e4 | 5.4e−7 | 376 | 9505 | 110 | 221 | 0 | |
| ABB | TMP-PCG | 2.8e−3 | 1.9e2 | 7.1e4 | 6.5e−15 | 211 | 11351 | 68 | 137 | 0 | prog. |
| | TMP-MINRES | 2.8e−3 | 1.9e2 | 7.1e4 | 9.3e−16 | 211 | 10253 | 68 | 137 | 0 | prog. |
| | SPG | 4.9e−3 | 1.6e2 | 7.1e4 | 4.5e−10 | 202 | 11187 | 65 | 131 | 0 | prog. |
| | TRON | 1.0e−4 | 8.0e2 | 7.1e4 | 1.2e−6 | 235 | 58716 | 75 | 151 | 0 | |
| | L-BFGS-B | 1.9e−4 | 2.2e2 | 7.1e4 | 2.7e−7 | 227 | 7548 | 73 | 147 | 0 | |
| $\text{ABB}_{\text{min1}}$ | TMP-PCG | 3.5e−4 | 2.1e2 | 7.1e4 | 2.5e−15 | 228 | 11776 | 70 | 141 | 0 | prog. |
| | TMP-MINRES | 3.7e−4 | 2.0e2 | 7.1e4 | 9.4e−16 | 220 | 10335 | 68 | 137 | 0 | prog. |
| | SPG | 1.7e−4 | 1.7e2 | 7.1e4 | 1.2e−8 | 219 | 11308 | 68 | 137 | 0 | |
| | TRON | 1.4e−3 | 4.9e2 | 7.1e4 | 3.6e−14 | 213 | 37906 | 66 | 133 | 0 | prog. |
| | L-BFGS-B | 3.7e−4 | 2.2e2 | 7.1e4 | 1.3e−7 | 220 | 7333 | 68 | 137 | 0 | prog. |
| $\text{ABB}_{\text{SS}}$ | TMP-PCG | 1.9e−4 | 2.0e2 | 7.1e4 | 7.7e−14 | 221 | 11486 | 71 | 143 | 0 | |
| | TMP-MINRES | 7.9e−4 | 1.7e2 | 7.1e4 | 7.8e−16 | 196 | 9608 | 63 | 127 | 0 | prog. |
| | SPG | 1.3e−4 | 1.8e2 | 7.1e4 | 2.0e−10 | 233 | 11579 | 75 | 151 | 0 | |
| | TRON | 1.9e−4 | 4.9e2 | 7.1e4 | 1.4e−14 | 219 | 37979 | 71 | 143 | 0 | |
| | L-BFGS-B | 1.9e−4 | 2.3e2 | 7.1e4 | 1.3e−7 | 233 | 7598 | 75 | 151 | 0 | |

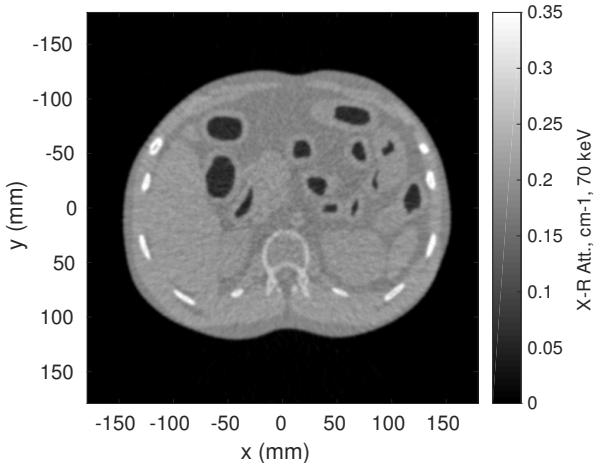
exceptions; mostly combinations involving SPG that are less accurate, stagnate or fail to converge.

Furthermore, the results highlight the effectiveness of the scaling (4.11), as the run times for TRON are similar to those for L-BFGS-B. Such results are encouraging and show the potential of specifically-tailored projection methods for large-scale problems.

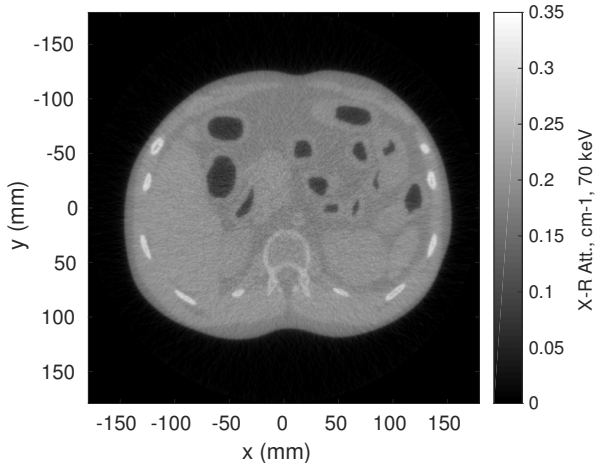
The discrepancy between the run times for TRON and SPG confirms that projections into the feasible set (4.17) dominate the cost of the projections on the active face (4.20). It is thus favorable to avoid taking too many projected gradient steps.

Our results show that, despite appearing prohibitive for large-scale problems, second-order methods can be competitive after all, provided that subproblems can be solved efficiently.

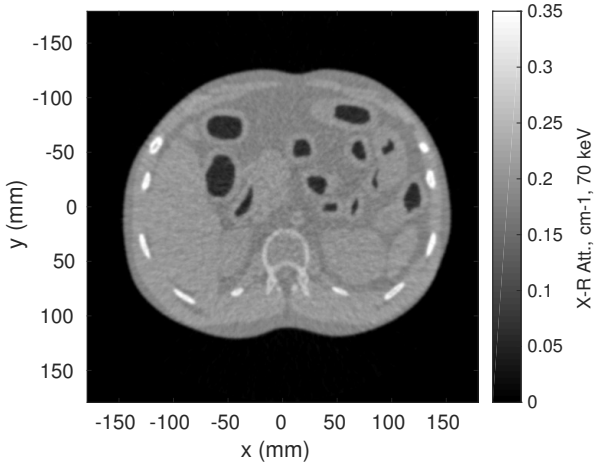
Figure 4.3 illustrates reconstructed images for the "best" candidate with each penalty function and each solver. Given that the results are fairly similar and that no combination stands out from the others, we pick those with the best overall performance: the combinations of SPG/ABB/TMP-MINRES and TRON/LSQR/TMP-MINRES. For the projection on the active face, we tend to favor LSQR and LSMR over their PCG and MINRES counterparts because of their improved numerical stability.



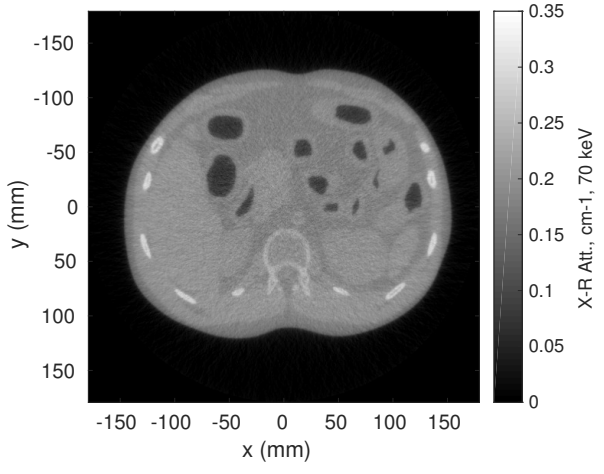

(a) TRON on (4.12), $\lambda = 0.1$



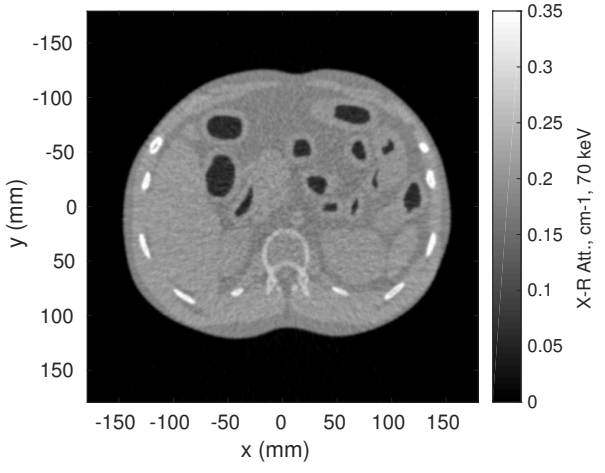

(b) TRON on (4.12), $\lambda = 0.1$



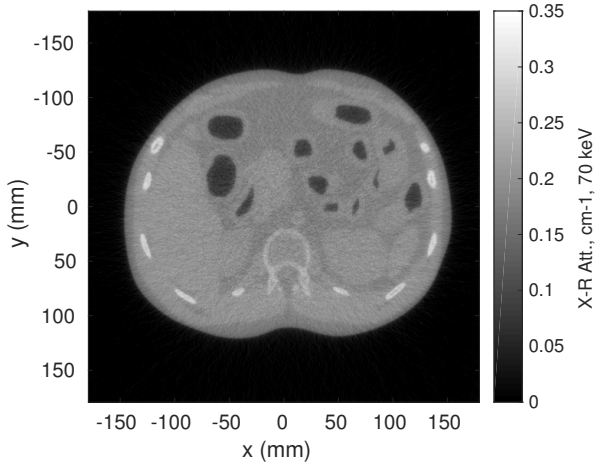

(c) SPG on (4.12), $\lambda = 0.1$




(d) SPG on (4.12), $\lambda = 0.1$




(e) L-BFGS-B on (4.10), $\lambda = 15$




(f) L-BFGS-B on (4.10), $\lambda = 25$

Figure 4.3 Reconstruction results using the $\mathscr{L}_2$ norm on the gradient of the object (left) and the $\mathscr{L}_2$ norm on the object (right). Problem (4.12) is posed in cylindrical coordinates while (4.10) is posed in Cartesian coordinates.

## 4.7 Discussion

We studied and designed factorization-free implementations for (4.12) by capitalizing on its fast Jacobian operator. This motivated us to develop efficient projection operations, and efficient implementations of projection-based active-set methods.

Our results not only show the effectiveness of specifically tailored projected methods but also that second-order methods on large-scale problems arising from image reconstruction can be viable.

Moreover, the radical reduction in memory requirements obtained using cylindrical coordinates might allow the application of iterative methods to 3D tomographic reconstruction on common computers. For that reason, we believe that the present work is a step towards applying iterative methods in clinical settings in the near future.

The impact of *inexact* projection must be assessed and it would be interesting to have theoretical results indicating how inexact projections are allowed to be. For the moment, we are content with using tight tolerances on the projection solvers, which ensure that infeasibility is insignificant in comparison to the accuracy of the iterates. Finally, we are considering factorization-free implementations of other approaches, including interior-point methods and proximal algorithms.

Implementations of our solvers are available in object-oriented MATLAB as part of the NLPLab optimization framework available at `https://bitbucket.org/maxmcl/nlplab`.

## 4.8 Appendix

### 4.8.1 Step Length Updates for Barzilai-Borwein Methods

In this section, we state the non-monotone Armijo linesearch proposed by Birgin *et al.* (2001) and Birgin et Martínez (2002) as 8. This line search uses quadratic interpolation to initialize $\lambda_{\text{temp}}$. Typical values are $\gamma = 1\text{e}{-4}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.9$ and $m = 10$.

**Algorithm 8** Safeguarded non-monotone Armijo line search

Given $0 < \sigma_1 < \sigma_2 < 1$, $\gamma \in (0,\ 1)$, $m \in \mathbb{N}_0^+$, $x_k \in \mathbb{R}^n$, $\nabla f(x_k)$ and a direction $d_k$,
set $f_{\max} = \max\{f(x_{k-j}) \mid 0 \leq j \leq \min(k,\ m-1)\}$
set $\lambda_k = 1$, $x_{k+1} = x_k + \lambda_k d_k$, $\delta = \nabla f(x_k)^\mathsf{T} d_k$
**while** $f(x_{k+1}) > f_{\max} + \gamma\lambda_k\delta$ **do**
    $\lambda_{\text{temp}} = -\lambda_k^2\delta / (2(f(x_{k+1}) - f(x_k) - \lambda_k\delta))$
    **si** $\sigma_1 \leq \lambda_{\text{temp}} \leq \sigma_2$ **then**
        set $\lambda_k = \lambda_{\text{temp}}$
    **else**
        set $\lambda_k = \frac{\lambda_k}{2}$
    **end si**
    $x_{k+1} = x_k + \lambda_k d_k$
**end while**
**return** $\lambda_k$, $x_{k+1}$

We now state the various step length update schemes that we consider for 5. Recall that at iteration $k$, $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$.

9, known as the *first* Barzilai-Borwein step length (BB1), gives the step used by Birgin *et al.* (2001). Typical values for $\alpha_{\max}$ and $\alpha_{\min}$ are 1.0e3 and 1.0e−3 respectively.

**Algorithm 9** Safeguarded first Barzilai-Borwein step (BB1)

Given the step length safeguards $\alpha_{\max} > \alpha_{\min} > 0$ and vectors $s_k$ and $y_k$
**si** $s_k^\mathsf{T} y_k < 0$ **then**
    **return** $\alpha_{\max}$
**else**
    **return** $\min\{\alpha_{\max},\ \max\{\alpha_{\min},\ s_k^\mathsf{T} s_k / s_k^\mathsf{T} y_k\}\}$
**end si**

We now state the various adaptative update rules that we consider. For a survey, we refer the reader to the works of di Serafino *et al.* (2017). 10 ($\text{ABB}_{\text{min1}}$), suggested by Frassoldati *et al.* (2008), determines whether the first or the smallest of the last $m_\alpha$ second Barzilai-Borwein step length should be used, given a threshold value $\tau$. The original ABB algorithm presented by Zhou *et al.* (2006) can be derived from 10 if we set $m_\alpha = 1$. Typical values of the parameters are $\tau = 0.8$ and $m_\alpha = 9$.

**Algorithm 10** Adaptive minimal Barzilai-Borwein step ($\text{ABB}_{\text{min1}}$)

Given $\tau$, the memory $m_\alpha$ and the inputs $s_k$ and $y_k$,
compute $\alpha^{(1)} = s_k^T s_k / s_k^T y_k$ and $\alpha^{(2)} = s_k^T y_k / y_k^T y_k$
**si** $\alpha^{(2)} < \tau\, \alpha^{(1)}$ **then**
  **return** $\min\{\alpha_j^{(2)} \mid j = \max(1,\, k - m_\alpha), \ldots,\, k\}$
**else**
  **return** $\alpha^{(1)}$
**end si**

Bonettini *et al.* (2009) propose 11 ($\text{ABB}_{\text{SS}}$), that is similar to 10, but uses a dynamic threshold $\tau$ and safeguards on $\alpha$, as in 9. Because (4.12) is already scaled, we set the scaling matrix as the identity in our implementation of 11. Typical values of the parameters are $\tau = 0.5$ and $m_\alpha = 2$.

**Algorithm 11** Safeguarded adaptive minimal B.-B. step with dynamic threshold ($\text{ABB}_{\text{SS}}$)

Given $\tau$, $\alpha_{\max} > \alpha_{\min} > 0$, the memory $m_\alpha$ and vectors $s_k$ and $y_k$,
**si** $s_k^T y_k \leq 0$ **then**
  $\alpha^{(1)} = \alpha_{\max}$
  $\alpha^{(2)} = \alpha_{\max}$
**else**
  $\alpha^{(1)} = \max\{\alpha_{\min},\, \min\{s_k^T s_k / s_k^T y_k,\, \alpha_{\max}\}\}$
  $\alpha^{(2)} = \max\{\alpha_{\min},\, \min\{s_k^T y_k / y_k^T y_k,\, \alpha_{\max}\}\}$
**end si**
**si** $\alpha^{(2)} \leq \tau\, \alpha^{(1)}$ **then**
  $\alpha = \min\{\alpha_j^{(2)} \mid j = \max(1,\, k - m_\alpha), \ldots,\, k\}$
  $\tau = 0.9 \cdot \tau$
**else**
  $\alpha = \alpha^{(1)}$
  $\tau = 1.1 \cdot \tau$
**end si**
**return** $\alpha$

### 4.8.2 TRON for Bounded Problems Compared to IPOPT

In this section, we validate our MATLAB implementation of TRON solver against IPOPT (Wächter et Biegler, 2006) on a selection of bound-constrained problems from the CUTE

library in AMPL format.[1] Our implementation of TRON accomodates (4.13) and we specialize it to bound constraints by supplying appropriate projection functions—see 4.3.1, 4.3.2 and 4.3.3. IPOPT is run with default parameters. (4.7) uses the same failure codes as in 4.6. Given the smaller sizes of the problems, we reduced the maximal allowed run time to 120 seconds and added a limit of 1e5 iterations, corresponding to the the error code "iter".

Table 4.7 – TRON vs. IPOPT on bound-constrained problems from CUTE.

| Problem | Solver | KKT | time | $f(x^\star)$ | #iter | $\#f$ | $\#g$ | $\#H$ | fail |
|---|---|---|---|---|---|---|---|---|---|
| 3pk | IPOPT | 3.10e−12 | 2e−2 | 1.72 | 11 | 12 | 12 | 11 | |
| | TRON | 6.68e−7 | 2e−1 | 1.72 | 111 | 111 | 111 | 220 | |
| allinit | IPOPT | 3.87e−15 | 4e−3 | 1.67e1 | 12 | 20 | 13 | 12 | |
| | TRON | 4.97e−9 | 1e2 | 1.67e1 | 18 | 44 | 9 | 44 | prog. |
| bdexp | IPOPT | 8.82e−11 | 1e−1 | 2.13e−8 | 21 | 22 | 22 | 21 | |
| | TRON | 4.45e−9 | 2e−1 | 2.13e−8 | 22 | 22 | 22 | 42 | |
| biggsb1 | IPOPT | 3.92e−16 | 2e−2 | 1.50e−2 | 20 | 21 | 21 | 20 | |
| | TRON | 2.73e−10 | 7e−1 | 1.50e−2 | 530 | 530 | 530 | 1058 | |
| bqpgabim | IPOPT | 1.39e−17 | 8e−3 | −3.79e−5 | 18 | 24 | 19 | 18 | |
| | TRON | 8.26e−11 | 3e−2 | −3.79e−5 | 8 | 8 | 8 | 14 | |
| bqpgasim | IPOPT | 5.02e−17 | 8e−3 | −5.52e−5 | 19 | 25 | 20 | 19 | |
| | TRON | 3.29e−10 | 2e−2 | −5.52e−5 | 26 | 84 | 13 | 66 | prog. |
| camel6 | IPOPT | 9.92e−16 | 8e−3 | −1.03 | 11 | 12 | 12 | 11 | |
| | TRON | 2.46e−14 | 1e−2 | −1.03 | 9 | 11 | 9 | 17 | |
| chenhark | IPOPT | 1.67e−15 | 3e−2 | −2.00 | 21 | 22 | 22 | 21 | |
| | TRON | 5.47e−8 | 5e1 | −2.00 | 3376 | 3434 | 3360 | 6767 | prog. |
| cvxbqp1 | IPOPT | 2.74e−12 | 6e−1 | 2.25e6 | 11 | 12 | 12 | 11 | |
| | TRON | 2.78e−15 | 1e−2 | 2.25e6 | 2 | 2 | 2 | 2 | |
| deconvb | IPOPT | 2.44e−9 | 1e1 | 3.04e−13 | 10000 | 47482 | 10001 | 10000 | iter. |
| | TRON | 3.36e−9 | 5e−2 | 1.33e−12 | 46 | 49 | 46 | 91 | |
| eg1 | IPOPT | 4.24e−16 | 4e−3 | −1.43 | 8 | 9 | 9 | 8 | |
| | TRON | 2.38e−8 | 1e2 | −1.13 | 19 | 51 | 7 | 48 | cpu. |
| explin | IPOPT | 4.34e−14 | 1e−2 | −7.24e5 | 19 | 20 | 20 | 19 | |
| | TRON | 2.33e−6 | 1e2 | −7.24e5 | 34 | 51 | 19 | 82 | cpu. |
| explin2 | IPOPT | 2.77e−14 | 2e−2 | −7.24e5 | 19 | 20 | 20 | 19 | |
| | TRON | 7.16e−5 | 1e2 | −7.24e5 | 31 | 48 | 14 | 77 | prog. |
| hadamals | IPOPT | 1.73e−14 | 2e−1 | 2.53e1 | 109 | 110 | 110 | 109 | |
| | TRON | 2.00e−9 | 2e−2 | 8.13e2 | 10 | 10 | 10 | 18 | |
| harkerp2 | IPOPT | 2.55e−15 | 2e−2 | −5.00e−1 | 17 | 18 | 18 | 17 | |
| | TRON | 0.00 | 1e−2 | −5.00e−1 | 3 | 3 | 3 | 4 | |
| hart6 | IPOPT | 1.90e−15 | 8e−3 | −3.32 | 9 | 15 | 10 | 9 | |
| | TRON | 4.59e−8 | 1e2 | −3.32 | 24 | 48 | 11 | 63 | prog. |
| hatflda | IPOPT | 2.23e−14 | 8e−3 | 9.52e−21 | 11 | 12 | 12 | 11 | |
| | TRON | 6.21e−13 | 2e−2 | 3.47e−25 | 30 | 30 | 30 | 58 | |
| hatfldb | IPOPT | 7.39e−15 | 1e−2 | 5.57e−3 | 11 | 12 | 12 | 11 | |
| | TRON | 2.00e−15 | 1e−2 | 5.57e−3 | 27 | 27 | 27 | 52 | |
| hatfldc | IPOPT | 1.16e−16 | 4e−3 | 6.44e−24 | 6 | 7 | 7 | 6 | |
| | TRON | 0.00 | 5e−3 | 0.00 | 6 | 6 | 6 | 10 | |

Continued on next page

1. https://github.com/mpf/Optimization-Test-Problems

Table 4.7 TRON vs. IPOPT on bound-constrained problems from CUTE – (cont'd)

| Problem | Solver | KKT | time | $f(x^\star)$ | #iter | $\#f$ | $\#g$ | $\#H$ | fail |
|---|---|---|---|---|---|---|---|---|---|
| himmelp1 | IPOPT | 5.79e−15 | 1e−2 | −6.21e1 | 13 | 21 | 14 | 13 | |
| | TRON | 1.09e−8 | 1e2 | −6.21e1 | 23 | 91 | 14 | 56 | prog. |
| hs110 | IPOPT | 3.68e−15 | 4e−3 | −4.58e1 | 7 | 8 | 8 | 7 | |
| | TRON | 8.44e−12 | 1e−2 | −4.58e1 | 18 | 18 | 18 | 34 | |
| hs3mod | IPOPT | 0.00 | 8e−3 | −9.99e−9 | 6 | 7 | 7 | 6 | |
| | TRON | 0.00 | 5e−3 | 0.00 | 5 | 5 | 5 | 8 | |
| logros | IPOPT | 4.18e−13 | 8e−2 | 0.00 | 76 | 679 | 77 | 76 | |
| | TRON | 2.56e−9 | 1e2 | 0.00 | 47 | 81 | 33 | 115 | cpu. |
| maxlika | IPOPT | 1.51e−12 | 2e−2 | 1.14e3 | 20 | 26 | 21 | 20 | |
| | TRON | 5.71e−11 | 4e−2 | 1.15e3 | 14 | 18 | 14 | 28 | |
| mccormck | IPOPT | 9.17e−16 | 4 | −4.57e4 | 18 | 237 | 19 | 18 | |
| | TRON | 4.69e−5 | 1e2 | −4.57e4 | 30 | 98 | 16 | 77 | cpu. |
| mdhole | IPOPT | 0.00 | 4e−2 | −9.99e−9 | 48 | 119 | 49 | 48 | |
| | TRON | 8.64e−18 | 1e−2 | 1.87e−37 | 15 | 21 | 15 | 30 | |
| ncvxbqp1 | IPOPT | 9.18e−13 | 3e1 | −1.99e10 | 261 | 262 | 262 | 261 | |
| | TRON | 5.67e−16 | 2e−2 | −1.99e10 | 2 | 2 | 2 | 2 | |
| ncvxbqp2 | IPOPT | 2.82e4 | 1e2 | −1.33e10 | 1161 | 1162 | 1162 | 1161 | xfail. |
| | TRON | 1.27e−2 | 1e2 | −1.33e10 | 32 | 59 | 13 | 81 | cpu. |
| ncvxbqp3 | IPOPT | 1.02e5 | 1e2 | −6.32e9 | 975 | 976 | 976 | 975 | xfail. |
| | TRON | 1.76e−3 | 1e2 | −6.56e9 | 29 | 117 | 14 | 72 | prog. |
| nonscomp | IPOPT | 2.51e−11 | 3e−1 | 1.96e−8 | 27 | 88 | 28 | 27 | |
| | TRON | 7.88e−7 | 1e−1 | 9.79e−15 | 12 | 12 | 12 | 22 | |
| obstclal | IPOPT | 2.73e−16 | 1e−2 | 1.40 | 15 | 16 | 16 | 15 | |
| | TRON | 2.23e−11 | 8e−3 | 1.40 | 11 | 13 | 11 | 21 | |
| obstclbl | IPOPT | 2.35e−16 | 4e−3 | 2.88 | 12 | 13 | 13 | 12 | |
| | TRON | 3.36e−9 | 1e2 | 2.88 | 20 | 91 | 8 | 50 | cpu. |
| obstclbu | IPOPT | 2.49e−16 | 8e−3 | 2.88 | 13 | 14 | 14 | 13 | |
| | TRON | 2.16e−11 | 6e−3 | 2.88 | 7 | 7 | 7 | 12 | |
| oslbqp | IPOPT | 9.09e−17 | 1e−2 | 6.25 | 17 | 18 | 18 | 17 | |
| | TRON | 0.00 | 4e−3 | 6.25 | 2 | 2 | 2 | 2 | |
| palmer1 | IPOPT | 1.77e−9 | 4e−1 | 1.18e4 | 766 | 1983 | 767 | 767 | prog. |
| | TRON | 1.87e−8 | 3e−2 | 2.82e4 | 61 | 61 | 61 | 120 | |
| palmer5a | IPOPT | 1.91e1 | 7 | 2.82e−2 | 10000 | 50184 | 10001 | 10000 | iter. |
| | TRON | 8.91e−4 | 5 | 5.86e−2 | 9802 | 10000 | 9802 | 19695 | prog. |
| palmer5b | IPOPT | 2.15e−11 | 4e−2 | 9.75e−3 | 80 | 203 | 81 | 80 | |
| | TRON | 1.63e−5 | 2e−1 | 1.50e−2 | 422 | 437 | 422 | 849 | |
| palmer5d | IPOPT | 1.55e−12 | 0 | 8.73e1 | 1 | 2 | 2 | 1 | |
| | TRON | 3.09e−8 | 4e−3 | 8.73e1 | 4 | 4 | 4 | 6 | |
| palmer5e | IPOPT | 6.24e−3 | 8 | 2.07e−2 | 10000 | 69390 | 10001 | 10000 | iter. |
| | TRON | 5.39e−4 | 6 | 3.87e−2 | 9967 | 10000 | 9967 | 19948 | prog. |
| palmer6a | IPOPT | 7.28e−11 | 1e−1 | 5.59e−2 | 288 | 701 | 289 | 288 | |
| | TRON | 1.78e−7 | 2e−1 | 5.59e−2 | 372 | 384 | 372 | 747 | |
| palmer6e | IPOPT | 7.06e−13 | 6e−2 | 2.24e−4 | 28 | 64 | 29 | 28 | |
| | TRON | 7.00e−7 | 1e−1 | 2.24e−4 | 177 | 207 | 177 | 367 | |
| palmer7e | IPOPT | 2.12e3 | 7 | 6.46 | 10000 | 40833 | 10001 | 10000 | iter. |
| | TRON | 3.06e−6 | 5e−1 | 1.02e1 | 944 | 991 | 944 | 1908 | |
| palmer8a | IPOPT | 4.32e−13 | 4e−2 | 7.40e−2 | 86 | 172 | 87 | 86 | |
| | TRON | 4.15e−7 | 4e−2 | 7.40e−2 | 75 | 75 | 75 | 148 | |
| palmer8e | IPOPT | 1.56e−11 | 1e−2 | 6.34e−3 | 28 | 44 | 29 | 28 | |
| | TRON | 9.79e−7 | 4e−2 | 6.34e−3 | 74 | 82 | 74 | 149 | |

Continued on next page

Table 4.7 TRON vs. IPOPT on bound-constrained problems from CUTE – (cont'd)

| Problem | Solver | KKT | time | $f(x^\star)$ | #iter | $\#f$ | $\#g$ | $\#H$ | fail |
|---|---|---|---|---|---|---|---|---|---|
| pentdi | IPOPT | 4.44e−16 | 2e−2 | −7.50e−1 | 18 | 19 | 19 | 18 | |
| | TRON | 0.00 | 4e−3 | −7.50e−1 | 2 | 2 | 2 | 2 | |
| probpenl | IPOPT | 2.62e−5 | 1e2 | −9.32e4 | 2548 | 2989 | 2549 | 2548 | xfail. |
| | TRON | 3.24e−6 | 1e−2 | 3.99e−7 | 2 | 2 | 2 | 2 | |
| pspdoc | IPOPT | 1.11e−16 | 8e−3 | 2.41 | 8 | 16 | 9 | 8 | |
| | TRON | 4.33e−14 | 5e−3 | 2.41 | 8 | 8 | 8 | 14 | |
| qr3dls | IPOPT | 3.80e−14 | 1e−1 | 1.34e−21 | 50 | 115 | 51 | 50 | |
| | TRON | 2.61e−10 | 4e−1 | 4.66e−18 | 92 | 109 | 92 | 190 | |
| qrtquad | IPOPT | 3.18e−11 | 2e−2 | −3.65e6 | 30 | 48 | 31 | 30 | |
| | TRON | 3.16e−6 | 1e2 | −3.65e6 | 62 | 178 | 50 | 151 | prog. |
| qudlin | IPOPT | 3.55e−15 | 2e−2 | −7.20e3 | 53 | 101 | 54 | 53 | |
| | TRON | 0.00 | 2e−3 | −7.20e3 | 2 | 2 | 2 | 2 | |
| s368 | IPOPT | 1.55e−16 | 1e−1 | 3.01e−20 | 7 | 8 | 8 | 7 | |
| | TRON | 0.00 | 9e−3 | 0.00 | 1 | 1 | 1 | 0 | |
| scon1dls | IPOPT | 4.83e−12 | 1 | 1.65e−16 | 458 | 2379 | 459 | 458 | |
| | TRON | 2.91e−4 | 1e2 | 8.06e−4 | 8830 | 9757 | 8830 | 18061 | cpu. |
| sim2bqp | IPOPT | 2.95e−18 | 4e−3 | −9.99e−9 | 8 | 9 | 9 | 8 | |
| | TRON | 0.00 | 3e−3 | 0.00 | 2 | 2 | 2 | 2 | |
| simbqp | IPOPT | 2.97e−18 | 8e−3 | −9.99e−9 | 8 | 9 | 9 | 8 | |
| | TRON | 0.00 | 3e−3 | 0.00 | 2 | 2 | 2 | 2 | |
| sineali | IPOPT | 1.70e−2 | 6 | −1.90e3 | 10000 | 26766 | 10001 | 10000 | iter. |
| | TRON | 4.44e−6 | 1e2 | −1.90e3 | 20 | 33 | 9 | 50 | cpu. |
| torsion-1 | IPOPT | 1.59e−16 | 1e−1 | −4.18e−1 | 17 | 18 | 18 | 17 | |
| | TRON | 8.52e−10 | 1e2 | −4.18e−1 | 35 | 123 | 24 | 81 | prog. |
| torsion-2 | IPOPT | 1.90e−16 | 2e−1 | −4.18e−1 | 18 | 19 | 19 | 18 | |
| | TRON | 9.13e−11 | 4e−1 | −4.18e−1 | 27 | 34 | 27 | 55 | |
| torsion-3 | IPOPT | 2.33e−16 | 2e−1 | −4.18e−1 | 18 | 19 | 19 | 18 | |
| | TRON | 2.75e−11 | 5e−1 | −4.18e−1 | 32 | 43 | 32 | 66 | |
| yfit | IPOPT | 4.20e−12 | 2e−2 | 6.67e−13 | 49 | 97 | 50 | 49 | |
| | TRON | 1.35e−8 | 3e−2 | 1.63e−12 | 63 | 81 | 63 | 132 | |

# CHAPITRE 5 DISCUSSION GÉNÉRALE

Afin de faciliter la discussion sur l'ensemble du mémoire, rappelons les faits saillants du problème que nous avons été amenés à considérer.

Dans le contexte de reconstruction d'images en tomographie par rayons X, nous considérons un algorithme de reconstruction itératif basé sur un estimateur de maximum a posteriori. Afin de produire une image, cette méthode requiert la résolution du problème (SP), qui est convexe et qui a la matrice de mise à l'échelle $C$ pour Jacobien des inégalités linéaires. Les particularités de ce problème sont sa taille élevée, qui rend impossible l'utilisation de matrices explicites, et le coût dispendieux d'une évaluation de la fonction objectif. À l'opposé, un produit avec $C$ est peu coûteux et nous pouvons également supposer que le problème (SP) est bien conditionné grâce à l'introduction de $C$.

Ces considérations nous amènent à concentrer nos efforts sur des solveurs sans factorisation qui pourraient tirer profit du faible coût des produits avec $C$. Dans l'article présenté à la section 4, nous justifions notre intérêt envers les méthodes d'ensemble actif basées sur des projections et démontrons leur performance. Alternativement, les algorithmes de points intérieurs, décrits à l'annexe E, sont des méthodes qui se prêtent bien aux problèmes de la forme (SP). Des travaux préliminaires ont été effectués à ce sujet, par l'entremise d'une variante du solveur PDCO de Saunders (2017). À première vue, la performance de ce dernier sur le problème (SP) semblait inférieure à celle des méthodes projetées. Pour une tolérance d'optimalité de 1e−8 et un temps d'exécution maximal de 15 minutes [1], les résultats obtenus pour les fonctions de pénalisation $\mathscr{L}_2$ sur le gradient de l'objet et sur l'objet, toutes deux avec $\lambda = 0.1$, sont présentés au tableau 5.1 [2]. Nous opposons deux formulations différentes des conditions de $\mathcal{KKT}$, respectivement nommées K2 et K35. Pour plus de détails, nous référons le lecteur à (Saunders, 2017).

1. Ce sont les mêmes paramètres que dans l'article présenté à la section 4.
2. Pour plus d'information sur les colonnes du tableau, le lecteur peut se rapporter à l'article.

| Formulation | Pénalité | KKT | temps [s] | $f(x^\star)$ | $\left\|\mu < 0\right\|$ | #P | #C | #iter | Sortie |
|---|---|---|---|---|---|---|---|---|---|
| K2 | Gradient | 4.5e1 | 1.1e3 | 8.3e9 | 0 | 2.6e3 | 6.6e3 | 9 | temps max. |
| | Objet | 1.4e3 | 9.1e2 | 3.4e11 | 0 | 2.1e3 | 5.2e3 | 5 | temps max. |
| K35 | Gradient | 4.8e2 | 9.8e2 | 1.3e11 | 0 | 2.0e3 | 4.9e3 | 5 | temps max. |
| | Objet | 1.1e2 | 1.0e3 | 2.5e10 | 0 | 2.3e3 | 5.7e3 | 9 | temps max. |

Tableau 5.1 Résultats de PDCO sur le problème (SP) pour des fonctions de pénalité $\mathscr{L}_2$ sur le gradient de l'objet et l'objet avec $\lambda = 0.1$

Par observation du tableau 5.1, on constate d'emblée qu'aucune des formulations ne converge pour le temps maximal alloué, contrairement à chacune des méthodes à base de projections considérées dans l'article. Une inspection plus minutieuse révèle que le nombre de produits avec la matrice de projection $P$ — notre principale métrique de performance — est supérieure d'un ordre de grandeur à celle des méthodes à base de projection. Nous pouvons donc supposer que cela explique les temps d'exécution considérablement plus élevés. D'ailleurs, des tests sur des images à plus faible résolution révèlent que PDCO converge vers la solution optimale, pourvu qu'elle ait suffisamment de temps. En raison de sa piètre performance initiale et notre intérêt envers les méthodes projetées, une investigation profonde n'a pas été menée quant aux différentes formulations et paramètres de PDCO. Il est possible que d'autres méthodes de points intérieurs sans factorisation soient applicables à notre problème et puissent mieux performer que PDCO. Ce sujet demeure matière à étude.

Finalement, le sujet de l'*inexactitude* des projections doit être abordé. Puisque les opérations de projection présentées dans l'article à la section 4 n'ont pas de *solutions analytiques*, des problèmes d'optimisation, ou encore des systèmes d'équations linéaires, doivent être résolus.

Lors de la mise en oeuvre, une certaine métrique représentant l'optimalité de la solution est choisie et nous tentons de la faire décroître en deçà d'une tolérance. Il est donc impossible d'avoir une solution *exacte* au sens numérique. Cela soulève une panoplie de questions : qu'arrive-t-il si la tolérance demandée ne peut être atteinte ? Quelle tolérance doit être exigée pour les projections ? Quelle est la relation entre la tolérance d'optimalité du problème de reconstruction et celle des problèmes de projection ? Comment l'inexactitude des projections affecte-elle le comportement des solveurs appliqués au problème de reconstruction ?

Hormis par une approche empirique, il ne semble pas y avoir de réponse définitive à ces interrogations. Nos conclusions ne sont donc pas forcément transférables à tout solveur qui contiendrait des projections inexactes. Il va sans dire qu'afin d'assurer la réalisabilité des solutions, i.e. le respect des contraintes, les projections doivent être résolues à une tolérance exigeante [3]. En pratique, nous pourrions imposer une tolérance sur les projections qui soit *significativement* plus exigeante que celle du problème de reconstruction, de sorte à ce que l'infaisabilité soit d'une ordre de grandeur négligeable par rapport à la précision de la solution. Soulignons également que des tolérances moins exigeantes signifient des temps d'exécution plus rapides, de sorte qu'il est possible de bénéficier d'un bon ajustement des tolérances. Advenant qu'une projection ne puisse être résolue sous le seuil de la tolérance demandée, nous traitons cela comme un échec et arrêtons l'algorithme de reconstruction. Finalement, dans le

3. Afin de lever toute ambiguïté, nous désignons une tolérance basse par "plus exigeante", e.g. $10^{-10}$, et une tolérance élevée par "moins exigeante", e.g. $10^{-3}$

cas de solveurs cherchant à identifier l'ensemble actif — tel que TRON —, il est primordial que les projections soient précises afin d'assurer l'identification correcte de l'ensemble actif. Toutes ces raisons nous amènent à exiger une tolérance sur les projections à un facteur d'au plus $10^{-3}$ de la tolérance utilisée pour le problème de reconstruction. Cette tolérance devrait également être ajustée en fonction de la nature de l'application et de l'importance de la faisabilité de la solution.

# CHAPITRE 6 CONCLUSION

## 6.1 Synthèse des travaux

Pour conclure sur nos travaux, nous avons démontré qu'il est possible de projeter efficacement dans l'ensemble réalisable de (SP) en bénéficiant du faible coût des produits avec $C$ et en traitant plutôt le dual du problème de projection. Dans le cadre de TRON, nous sommes amenés à considérer deux opérations de projection supplémentaires. La première, qui implique de projeter dans la face active, possède une solution analytique et est résolue efficacement par le biais d'une méthode de Krylov. En ce qui concerne la seconde, que nous désignons comme "projection mixte", nous avons démontré qu'elle consiste en un cas particulier de la projection dans l'ensemble réalisable et est au plus aussi difficile que celle-ci. En conséquence, nous sommes parvenus à développer des méthodes de contraintes actives à base de projections qui sont performantes sur le problème (SP), notamment une adaptation de TRON et le gradient projeté spectral.

En dépit du fait que le problème (SP) soit de grande taille et que TRON soit une méthode d'ordre deux, nos résultats prouvent que ce dernier est supérieur au gradient projeté sur (SP). Cela va à l'encontre du consensus dans le domaine, à savoir que les méthodes d'ordre deux sont trop coûteuses pour être utilisées sur des problèmes de tailles imposantes et denses. Nous croyons que cela démontre les avantages que peuvent avoir des méthodes d'optimisation développées sur mesure, particulièrement dans le cas de projections sur l'ensemble réalisable. Cela illustre l'importance d'exploiter les particularités d'un problème plutôt que d'utiliser des implémentations génériques.

Finalement, en dépit de la complexité supérieure de notre problème, nous avons réussi à demeurer compétitifs avec L-BFGS-B sur le problème (OP) en coordonnées cartésiennes. Nous obtenons donc une performance similaire tout en consommant drastiquement moins de mémoire. Nous sommes d'avis que ces avancées sont propices à l'intégration de méthodes itératives statistiques en clinique dans un proche futur.

## 6.2 Limitations de la solution proposée

D'emblée, on peut identifier les limitations les plus évidentes de nos solveurs : leur performance dépend largement des opérations de projections qui ont été développées. En conséquence, ils ne sont pas réellement transférables à d'autres applications, ou du moins leur efficacité n'est pas garantie. Une autre limitation est plus subtile, soit l'inexactitude des pro-

jections. Ce sujet demeure matière à étude, mais il serait intéressant d'avoir des résultats théoriques afin d'appuyer nos algorithmes. Bien qu'il semble requis d'avoir des projections exactes, nous pourrions bénéficier de projections inexactes qui requièrent un nombre moins élevé de calculs.

D'un point de vue plus conceptuel, nous pouvons souligner que nous traitons exclusivement une modélisation *monochromatique* du phénomène. Il serait intéressant de considérer l'approche polychromatique proposée par Hamelin (2009) afin de vérifier si la performance de nos solveurs est maintenue. De plus, nous pourrions considérer de vrais cas cliniques où la réduction des artefacts métalliques est cruciale afin de pouvoir correctement interpréter les images.

## 6.3 Améliorations futures

Maintenant que les avantages des méthodes à base de projections ont été démontrés sur le problème (SP), nous pourrions chercher à implémenter notre adaptation de TRON dans un langage de bas niveau, comme le C++. Bien que le coût de la résolution du problème (SP) soit dominé par les produits avec la matrice de projection et les projections sur l'ensemble réalisable, l'utilisation d'un tel langage entraînerait définitivement des gains au niveau du temps d'exécution. Plutôt que d'implémenter notre adaptation de TRON, nous pourrions appliquer le même genre de raisonnement à un solveur plus répandu qui possède déjà une implémentation en un langage de bas niveau, tel que L-BFGS-B. D'ailleurs, il serait intéressant d'investiguer si l'utilisation d'un hessien quasi-Newton pourrait être bénéfique sur le problème (SP). À court terme, il semble réalisable de substituer le hessien exact par un hessien quasi-Newton dans notre implémentation de TRON afin de valider ou d'invalider cette hypothèse.

Pour ce qui est de la modélisation, soulignons que l'impact de la pénalisation (1.18) et de la formulation du problème de reconstruction (OP) où $\Delta_N = \operatorname{diag}(n)$ n'ont pas été investigués. Il serait intéressant de comparer toutes les formulations dont nous disposons afin d'identifier leurs bénéfices.

Dans un autre ordre d'idée, rappelons notre intérêt envers les méthodes de points intérieurs pour ce qui a trait à la résolution du problème (SP). La variante de PDCO (Saunders, 2017) que nous avons considéré devrait être étudiée plus rigoureusement afin de déterminer s'il est avantageux de l'employer sur (SP). Alternativement, si une méthode de points intérieurs sans factorisation implémentée dans un language de bas niveau était disponible, il serait définitivement intéressant de tenter de l'appliquer à (SP). Soulignons également notre intérêt

envers les méthodes proximales, qui ne sont pas décrites dans cet ouvrage, mais qui sont très en vogue dernièrement dans le domaine de l'optimisation (Parikh et Boyd, 2014).

# RÉFÉRENCES


Barzilai, Jonathan and Borwein, Jonathan M. (1988). Two-point step size gradient methods. *IMA J. Numer. Anal.*, *8*(1), 141–148.

Beister, M. and Kolditz, D. and Kalender, W.A. (2012). Iterative reconstruction methods in X-ray CT. *Physica Medica*, *28*(2), 94–108.

Bertsekas, Dimitri P. (1982). Projected Newton Methods for Optimization Problems with Simple Constraints. *SIAM J. Control Optim.*, *20*(2), 221–246.

Birgin, Ernesto G. and Martínez, José M. (2002). Large-Scale Active-Set Box-Constrained Optimization Method with Spectral Projected Gradients. *Sci. (80-. ).*, *23*(1), 101–125.

Birgin, Ernesto G. and Martínez, José M. and Raydan, Marcos (2001). Algorithm 813 : SPG—software for convex-constrained optimization. *ACM Trans. Math. Softw.*, *27*(3), 340–349.

Bonettini, Silvia and Zanella, Riccardo and Zanni, Luca (2009). A scaled gradient projection method for constrained image deblurring. *Inverse Probl.*, *25*(1), 015002.

Bouman, Charles and Sauer, Ken (1993). A Generalized Gaussian Image Model for Edge-Preserving MAP Estimation. *IEEE Trans. Image Process.*, *2*(3), 296–310.

Boyd, Stephen and Vandenberghe, Lieven (2010). *Convex Optimization*, vol. 25. Cambridge University Press, New York, NY, USA.

Brenner, David J. and Hall, Eric J. (2007). Computed tomography–an increasing source of radiation exposure. *N. Engl. J. Med.*, *357*(22), 2277–2284.

Byrd, Richard H. and Lu, Peihuang and Nocedal, Jorge and Zhu, Ciyou (1995). A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM J. Sci. Comput.*, *16*(5), 1190–1208.

Byrd, Richard H. and Nocedal, Jorge and Waltz, Richard A. (2006). Knitro : An Integrated Package for Nonlinear Optimization. *Energy*, *83*, 1–25.

Conn, Andrew R. and Gould, Nicholas I. M. and Toint, Philippe L. (2000). Trust-region methods. *Book*, *MPS/SIAM S*, xx+959.

Cooley, James W. and Tukey, John W. (1965). An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comput.*, *19*(90), 297–301.

Dai, Yu Hong and Fletcher, Roger (2005). Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.*, *100*(1), 21–47.

di Serafino, Daniela and Ruggiero, Valeria and Toraldo, Gerardo and Zanni, Luca (2017). On the steplength selection in gradient methods for unconstrained optimization. Rapport technique, Optimization Online.

Fessler, Jeffrey A. (2000). Statistical image reconstruction methods for transmission tomography.

Fong, David C.-L. and Saunders, Michael A. (2011). LSMR : An Iterative Algorithm for Sparse Least-Squares Problems. *SIAM J. Sci. Comput.*, *33*(5), 2950–2971.

Frassoldati, Giacomo and Zanni, Luca and Zanghirati, Gaetano (2008). New adaptive stepsize selections in gradient methods. *J. Ind. Manag. Optim.*, *4*(2), 299–312.

Gafni, Eli M. and Bertsekas, Dimitri P. (1984). Two-Metric Projection Methods for Constrained Optimization. *SIAM J. Control Optim.*, *22*(6), 936–964.

Giovannelli, Jean-François and Idier, Jérôme (2013). *Méthodes d'inversion appliquées au traitement du signal et de l'image*. Hermès - Lavoisier.

Golkar, Masha A. (2013). *Fast Iterative Reconstruction in X-ray Tomography Using Polar Coordinates*. Mémoire de maîtrise, École Polytechnique de Montréal.

Gould, Nicholas I. M. and Hribar, Mary E. and Nocedal, Jorge (2001). On the Solution of Equality Constrained Quadratic Programming Problems Arising in Optimization. *SIAM J. Sci. Comput.*, *23*(4), 1376–1395.

Gould, Nicholas I. M. and Orban, Dominique and Rees, Tyrone (2013). Projected Krylov methods for saddle-point systems. *Cah. du GERAD G-2013-23, GERAD*, 1–26.

Goussard, Yves and Golkar, Mahsa A. and Wagner, Adrien and Voorons, Matthieu (2013). Cylindrical coordinate representation for statistical 3D CT reconstruction. *12th Int. Meet. Fully Three-Dimensional Image Reconstr. Radiol. Nucl. Med.*, 138–141.

Hamelin, Benoit (2009). *Accélération d'une approche régularisée de reconstruction en tomographie à rayons X avec réduction des artéfacts métalliques*. Thèse de doctorat, École Polytechnique de Montréal.

Herman, Gabor (2009). *Fundamentals of Computerized Tomography : Image Reconstruction from Projections*. Springer Publishing Company, Incorporated, seconde édition.

Herman, Gabor T. and Rowland, Stuart W. (1973). Three methods for reconstructing objects from X-rays : A comparative study. *Comput. Graph. Image Process.*, *2*, 151–178.

Hestenes, Magnus R. and Stiefel, Eduard (1952). Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand. (1934).*, *49*(6), 409–436.

Lin, Chih-Jen and Moré, Jorge J. (1999). Newton's method for large bound-constrained optimization problems. *SIAM J. Optim.*, *9*(4), 1100–1127.

Luenberger, David G. and Ye, Yinyu (2008). *Linear and Nonlinear Programming*. Springer.

Navidi, William (2010). *Statistics for Engineers and Scientists*. McGraw-Hill Higher Education.

Nocedal, Jorge and Wright, Stephen (2000). *Numerical Optimization*. Springer.

Paige, Christopher C. and Saunders, Michael A. (1975). Solution of Sparse Indefinite Systems of Linear Equations. *SIAM J. Numer. Anal.*, *12*(4), 617–629.

Paige, Christopher C. and Saunders, Michael A. (1982). LSQR : An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Trans. Math. Softw.*, *8*(1), 43–71.

Pan, Xiaochuan and Sidky, Emil Y. and Vannier, Michael (2009). Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction ? *Inverse Probl.*, *25*(12).

Parikh, Neal and Boyd, Stephen (2014). Proximal Algorithms. *Found. Trends® Optim.*, *1*(3), 127–239.

Petersen, Kaare B. and Pedersen, Michael S. (2007). The Matrix Cookbook.

Prince, Jerry L. and Links, Jonathan M. (2007). *Medical Imaging Signals and Systems*. Pearson.

Sauer, Ken and Bouman, Charles (1993). A Local Update Strategy for Iterative Reconstruction from Projections. *IEEE Trans. Signal Process.*, *41*(2), 534–548.

Saunders, Michael (2017). PDCO : Primal-Dual Interior Methods. `http://stanford.edu/class/msande318/notes/notes09-PDCO.pdf`.

Schmidt, Mark and Kim, Dongmin and Sra, Suvrit (2011). Projected Newton-type Methods in Machine Learning. *Optim. Mach. Learn.*, MIT Press, Cambridge, MA, USA. 305–330.

Segars, Williams P. and Mahesh, Mahadevappa and Beck, Thomas J. and Frey, Eric C. and Tsui, Benjamin M. W. (2008). Realistic CT simulation using the 4D XCAT phantom. *Med. Phys.*, *35*(8), 3800–3808.

Steihaug, Trond (1983). The Conjugate Gradient Method and Trust Regions in Large Scale Optimization. *SIAM J. Numer. Anal.*, *20*(3), 626–637.

Thibaudeau, Christian and Leroux, Jean-Daniel and Fontaine, Réjean and Lecomte, Roger (2013). Fully 3D iterative CT reconstruction using polar coordinates. *Med. Phys.*, *40*(11), 111904.

Trefethen, Lloyd N. and Bau III, David (1997). Numerical linear algebra. *Numer. Linear Algebr. with Appl.*, *12*, 361.

Wächter, Andreas and Biegler, Lorenz T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.*, *106*(1), 25–57.

Wirgin, Armand (2004). The inverse crime. Working paper or preprint.

Zeng, Gengsheng L. and Gullberg, Grant T. (1993). A Ray-driven Backprojector For Backprojection Filtering And Filtered Backprojection Algorithms. *1993 IEEE Conf. Rec. Nucl. Sci. Symp. Med. Imaging Conf.* 1199–1201.

Zhou, Bin and Gao, Li and Dai, Yu-Hong (2006). Gradient Methods with Adaptive Step-Sizes. *Comput. Optim. Appl.*, *35*(1), 69–86.

Zhu, Ciyou and Byrd, Richard H. and Lu, Peihuang and Nocedal, Jorge (1997). Algorithm 778 : L-BFGS-B : Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, *23*(4), 550–560.

# ANNEXE A FORMALISME EN COORDONNÉES CYLINDRIQUES

La discrétisation en coordonnées cylindriques adoptée par Goussard *et al.* (2013) est basée principalement sur le travail de Thibaudeau *et al.* (2013). Dans le contexte du calcul d'une matrice de projection, les coordonnées cylindriques permettent de bénéficier d'invariances en symétrie dûes à la géométrie cylindrique des tomographes.

Afin d'assurer cette symétrie, assumons que l'axe de rotation du tomographe est superposé à celui du système de coordonnées et que les voxels azimuthaux sont discrétisés en un multiple entier du nombre de projections et sont espacés également. La première exigence est pratiquement toujours respectée, en vertu de la géométrie des tomographes modernes. La deuxième peut être assurée par notre choix du nombre de voxels et permet d'indexer les voxels azimuthaux selon un incrément angulaire.

Le tracé de rayons proposé par Thibaudeau *et al.* (2013) traite les trois directions cylindriques, soit azimuthale, radiale et axiale, de manière séparée. En conséquence, les longueurs d'intersection entre un rayon, i.e. une droite reliant la source et un détecteur, et les frontières d'un voxel sont obtenues par la résolution de trois sous-problèmes indépendants. Les discrétisations azimuthales, radiales et axiales utilisées par Thibaudeau *et al.* (2013) sont présentées à la figure A.1. Par observation des intersections dans le plan transverse, Thibaudeau *et al.* (2013) dénotent que les longueurs d'intersection entre les rayons et les voxels sont les mêmes pour tout angle de rotation de la source et des détecteurs. En conséquence, il suffit de calculer l'opérateur de projection pour un seul angle pour l'obtenir à tout angle. Dans notre cas, puisque l'opérateur de projection est représenté sous forme matricielle, cela se manifeste par une matrice de projection *bloc-circulante*.

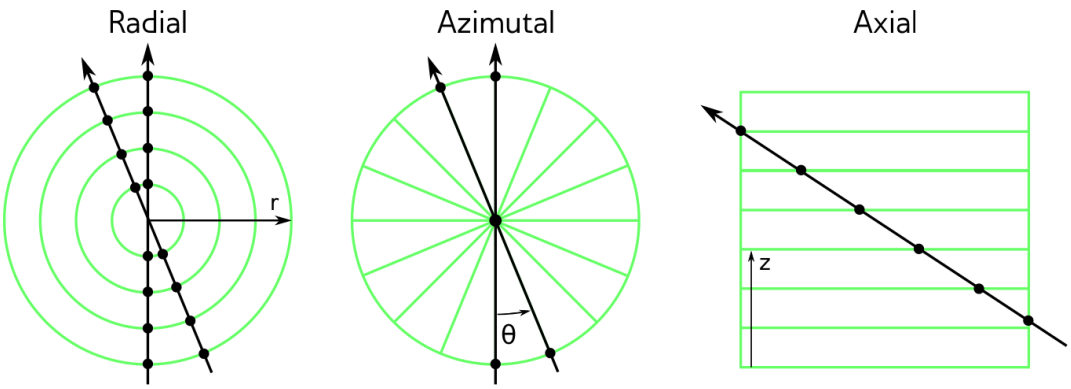



Figure A.1 Tracé de rayons en coordonnées cylindriques, adapté de Thibaudeau *et al.* (2013)

Sans perdre de généralité, on peut définir un opérateur $P_0$, représentant la matrice de projection *partielle* pour un angle de référence $\theta_0$. Goussard *et al.* (2013) dénotent que cette matrice est relativement creuse et de faible dimension, de sorte qu'il est facile de la stocker dans la mémoire. Dans le contexte d'une discrétisation en coordonnées cylindriques, Goussard *et al.* (2013) utilisent une approche *selon le rayon* (*ray driven*) [1], pour calculer $P_0$ et démontrent qu'il est possible d'obtenir la matrice de projection complète en appliquant l'opérateur de translation circulaire à $P_0$ :

$$P = \left[ P_0^{\mathsf{T}} \;\middle|\; (P_0 S_\mu^{-1})^{\mathsf{T}} \;\middle|\; \dots \;\middle|\; (P_0 S_\mu^{-n_\theta+1})^{\mathsf{T}} \right]^{\mathsf{T}}$$

où $P \in \mathbb{R}^{n_{\text{meas}} \times n_{\text{vox}}}$ et $S_\mu^{-i}$ représente l'opérateur de translation circulaire correspondant à une rotation de $-\theta_i$. En d'autres mots, on applique une *rotation* sur l'objet afin de le ramener à $\theta_0$ avant de lui appliquer l'opérateur de projection $P_0$.

Bien que l'utilisation d'une approche basée sur la rotation diminue fortement le stockage mémoire requis pour l'opérateur de projection $P$, ces rotations peuvent introduire des erreurs d'interpolation. Un raisonnement similaire est effectué par Goussard *et al.* (2013) pour l'opérateur de *rétroprojection*, soit $P^{\mathsf{T}}$.

Pour ce qui est de la matrice de différences premières utilisées dans les fonctions de pénalisation (1.17) et (1.18), Golkar (2013) définit $\{D_r,\ D_\theta,\ D_z\}$, de sorte à obtenir une sous-matrice de différences finies pour chaque direction de notre système de coordonnées. Le lecteur peut se rapporter à (Golkar, 2013) pour une représentation explicite des sous-matrices en question. Il est important de noter que ce sont généralement des matrices bidiagonales ou tridiagonales, dont la somme des lignes sont nulles.

1. Le lecteur peut se référer à (Zeng et Gullberg, 1993) pour plus de détails sur les approches dites *ray driven*.

# ANNEXE B PROPRIÉTÉS UTILES

## Décomposition spectrale des matrices bloc-circulantes

Toute matrice bloc-circulante $P$ peut s'exprimer sous la forme suivante — voir, e.g., (Petersen et Pedersen, 2007) :

$$P = \tfrac{1}{n}\, \mathscr{F}_n^\star\, \Pi\, \mathscr{F}_n, \tag{B.1}$$

où $\Pi$ est bloc-diagonale et où $\mathscr{F}_n$ est l'opérateur transformée de Fourier discret, avec $n$ le nombre de lignes de $\Pi$.

## Opérateur de transformée de Fourier

Au cours de cet ouvrage, nous utilisons la notation :

$$(\mathscr{F}_n)_{j,k} = \mathrm{e}^{2\pi \mathrm{i} jk},$$

pour désigner un élément de $\mathscr{F}_n$, l'opérateur de transformée de Fourier discret. L'opérateur de transformée de Fourier est unitaire à une constante près et on peut énoncer les identités suivantes — voir, e.g., (Petersen et Pedersen, 2007) :

$$\begin{aligned} \mathscr{F}_n^\star &= n\, \mathscr{F}_n^{-1}, \\ I &= n\, \mathscr{F}_n^{-\star}\, \mathscr{F}_n^{-1}, \\ \mathscr{F}_n\, \mathscr{F}_n^\star &= nI, \end{aligned} \tag{B.2}$$

où $(\cdot)^\star$ dénote la transposée conjuguée.

# ANNEXE C DÉRIVATION DE LA MATRICE DIAGONALE DANS LE DOMAINE DE FOURIER

Afin d'accélérer la convergence des solveurs lors de la résolution de (OP) en coordonnées cylindriques, Golkar (2013) a développé la matrice de *mise à l'échelle* $C$ (1.19). Elle est appliquée au problème (OP) par le changement de variable (1.20) et permet de diminuer l'étalement spectral du hessien du problème. Cette matrice a été retenue pour sa performance et sa facilité a être obtenue, comparativement à diverses matrices étudiées par Golkar (2013). On réfère à la matrice $C$ comme une matrice *diagonale dans le domaine de Fourier*, car elle est obtenue en prenant la diagonale de l'inverse du hessien de (OP) sous forme bloc-diagonale. Au cours de cette section, nous détaillons exactement de quelle façon $C$ est calculée pour le problème (OP).

Une fonction de pénalisation devra donc être définie pour le problème (OP) avant de pouvoir calculer la matrice $C$, car le hessien de la fonction objectif dépend de celle-ci. La forme générale du hessien de la fonction objectif $f(\mu)$ du problème (OP) est la suivante :

$$\nabla^2 f(\mu) = P^\mathsf{T} P + \lambda \nabla^2 \phi(\mu). \tag{C.1}$$

On dénote que puisque $P$ est bloc-circulant, alors $P^\mathsf{T} P$ est également bloc-circulant. Afin de préserver la convexité de la fonction objectif et de pouvoir bloc-diagonaliser $\nabla^2 f(\mu)$, nous imposons une fonction de pénalité non-linéaire convexe à hessien bloc-circulant. Dans le cas où le hessien ne serait pas bloc-circulant, on approxime $\phi(\mu)$ par un développement de Taylor d'ordre deux autour de $\mu_0$, de sorte qu'on obtient :

$$\nabla^2 f(\mu) \approx P^\mathsf{T} P + \lambda \nabla^2 \phi(\mu_0),$$

où on assume que $\phi(\mu)$ est choisie de sorte que $\nabla^2 \phi(\mu_0)$ soit bloc-circulant. On peut donc utiliser la propriété (B.1) et la propriété (B.2) afin d'écrire (C.1) sous forme bloc-diagonale :

$$\begin{aligned} P^\mathsf{T} P + \lambda \nabla^2 \phi(\mu_0) &= \left(\tfrac{1}{n} \mathscr{F}_n^\star \, \Pi_P \, \mathscr{F}_n\right)^\star \left(\tfrac{1}{n} \mathscr{F}_n^\star \, \Pi_P \, \mathscr{F}_n\right) + \left(\tfrac{1}{n} \mathscr{F}_n^\star \, \Pi_D \, \mathscr{F}_n\right), \\ &= \tfrac{1}{n} \mathscr{F}_n^\star \, \Pi_P^\star \Pi_P \, \mathscr{F}_n + \tfrac{1}{n} \mathscr{F}_n^\star \, \Pi_D \, \mathscr{F}_n, \\ &= \tfrac{1}{n} \mathscr{F}_n^\star \left(\Pi_P^\star \Pi_P + \Pi_D\right) \mathscr{F}_n, \\ &= \tfrac{1}{n} \mathscr{F}_n^\star \, \Pi \, \mathscr{F}_n \, . \end{aligned} \tag{C.2}$$

Puisque le produit d'une matrice bloc-diagonale transposée avec elle même est toujours bloc-

diagonal, et qu'une somme de matrices bloc-diagonales est encore bloc-diagonale, on peut réécrire le système en n'introduisant qu'une seule matrice bloc-diagonale, notée $\Pi$. L'inverse de notre approximation de la matrice hessienne s'écrit donc :

$$\begin{aligned}\left(P^{\mathsf{T}}P + \lambda\nabla^2\phi(\mu_0)\right)^{-1} &= \tfrac{1}{n}\left(\mathscr{F}_n^{\star}\,\Pi\,\mathscr{F}_n\right)^{-1}, \\ &= \tfrac{1}{n}\,\mathscr{F}_n^{-1}\,\Pi^{-1}\left(\mathscr{F}_n^{\star}\right)^{-1}, \\ &= \tfrac{1}{n}\,\mathscr{F}_n^{\star}\,\Pi^{-1}\,\mathscr{F}_n\,.\end{aligned} \tag{C.3}$$

L'objectif de la mise à l'échelle est de rapprocher les valeurs propres du hessien de notre problème, de sorte qu'on cherche une matrice $C$ facilement inversible telle que :

$$C^{\star}C \approx \left(P^{\mathsf{T}}P + \nabla^2\phi(\mu_0)\right)^{-1}. \tag{C.4}$$

Par observation de (C.3), on peut conclure que

$$C^{\star}C = \frac{1}{n}\,\mathscr{F}_n^{\star}\,\Pi^{-1}\,\mathscr{F}_n,$$

et l'approche retenue par Golkar (2013) implique simplement d'extraire la diagonale de $\Pi$, de sorte qu'on obtient :

$$C^{\star}C \approx \tfrac{1}{n}\,\mathscr{F}_n^{\star}\,\mathrm{diag}\left(\Pi\right)^{-1}\mathscr{F}_n\,.$$

Il est donc facile de calculer $\mathrm{diag}\left(\Pi\right)^{-1}$, car cela correspond à l'inverse de tous les éléments sur la diagonale de $\Pi$. De plus, on peut introduire $\mathrm{diag}\left(\Pi\right)^{-1/2}$ afin de réécrire :

$$\begin{aligned}C^{\star}C &= \tfrac{1}{n}\,\mathscr{F}_n^{\star}\,\mathrm{diag}\left(\Pi\right)^{-1/2}\mathrm{diag}\left(\Pi\right)^{-1/2}\mathscr{F}_n, \\ &= \tfrac{1}{n}\,\mathscr{F}_n^{\star}\,\mathrm{diag}\left(\Pi\right)^{-1/2}\mathscr{F}_n^{-\star}\,\mathscr{F}_n^{-1}\,\mathrm{diag}\left(\Pi\right)^{-1/2}\mathscr{F}_n, \\ &= \left(\mathscr{F}_n^{-1}\,\mathrm{diag}\left(\Pi\right)^{-1/2}\mathscr{F}_n\right)^{\star}\left(\mathscr{F}_n^{-1}\,\mathrm{diag}\left(\Pi\right)^{-1/2}\mathscr{F}_n\right).\end{aligned}$$

Par observation de l'équation précédente et par (B.2), on peut identifier la matrice $C$ :

$$C = \tfrac{1}{n}\,\mathscr{F}_n^{\star}\,\mathrm{diag}\left(\Pi\right)^{-1/2}\mathscr{F}_n, \tag{C.5}$$

que l'on réécrit plus simplement en définissant $\Delta = \mathrm{diag}\left(\Pi\right)^{-1/2}$, tel qu'à l'équation (1.19). Il est important de noter que nous introduisons une transformée de Fourier additionnelle afin d'éviter que $C$ soit de nature complexe, i.e. nous voulons repasser dans le domaine spatial. Un choix équivalent en matière de "conditionnement" est $C = \frac{1}{\sqrt{n}}\mathrm{diag}\left(\Pi\right)^{-1/2}\mathscr{F}_n$, mais est inévitablement complexe, i.e. un produit $Cx$ sera dans le domaine fréquentiel.

# ANNEXE D MODÉLISATION DES COMPTES DE PHOTONS PAR UNE LOI NORMALE

Soit $\mathcal{N}(m, \Sigma)$ une loi normale de moyenne $m$ et de matrice de covariance $\Sigma$. Considérons un bruit gaussien représenté par la variable aléatoire $B$, de réalisation $b$, de moyenne nulle et de covariance $\Delta_N^{-1}$ :

$$B \sim \mathcal{N}(0, \Delta_N^{-1}),$$

où $\Delta_N = \text{diag}\,(n))$, la matrice diagonale définie à la section 1.2. Nous pouvons définir le sinogramme *bruité* $y = P\mu - b$ en introduisant le bruit gaussien précédent dans (1.13). Par la définition de la fonction de densité de probabilité de $\mathcal{N}(0, \Delta_N^{-1})$ — voir, e.g., (Navidi, 2010) :

$$f(b) = \frac{\exp\left(-\frac{1}{2} b^\mathsf{T} \Delta_N b\right)}{\sqrt{|2\pi\Delta_N^{-1}|}},$$

on peut réécrire la fonction de probabilité conditionnelle :

$$p(y \,|\, \mu) = \frac{\exp\left(-\frac{1}{2}(P\mu - y)^\mathsf{T} \Delta_N (P\mu - y)\right)}{\sqrt{|2\pi\Delta_N^{-1}|}}.$$

Dans le cadre du maximum de vraisemblance (ML), on cherche la valeur la plus probable de cette distribution, sous contrainte de positivité par la définition de $\mu$, de sorte qu'on obtient :

$$\hat{\mu}_{\text{ML}} = \underset{\mu \geq 0}{\text{argmax}} \;\; p(y \,|\, \mu).$$

En considérant le cadre de la nég-log-vraisemblance utilisé dans 1.2, on peut réécrire l'expression précédente sous la forme :

$$\hat{\mu}_{\text{ML}} = \underset{\mu \geq 0}{\text{argmin}} \;\; \tfrac{1}{2}(P\mu - y)^\mathsf{T} \Delta_N (P\mu - y) + \ln\left(\sqrt{|2\pi\Delta_N^{-1}|}\right).$$

En éliminant les termes qui ne dépendent pas de $\mu$ et en réarrangeant l'expression, on obtient :

$$\hat{\mu}_{\text{ML}} = \underset{\mu \geq 0}{\text{argmin}} \;\; \tfrac{1}{2}\left\|P\mu - y\right\|_{\Delta_N}^2,$$

ce qui correspond au résultat obtenu en effectuant un développement d'ordre deux sur la distribution poissonnienne (1.15).

# ANNEXE E MÉTHODE DES POINTS INTÉRIEURS

Les méthodes dites de points intérieurs sont une famille de solveurs permettant de générer des itérés strictement à l'intérieur des contraintes d'inégalité du problème. Elles reposent sur des stratégies fortement similaires à celles du Lagrangien augmenté, à savoir que des variables d'écart sont introduites afin de transformer les inégalités en égalités. Au cours de cette section, nous présentons une méthode *primale-duale* générique, basée sur (Nocedal et Wright, 2000, chapitre 18). Les implémentations les plus reconnues des méthodes de points intérieurs sont probablement IPOPT de Wächter et Biegler (2006), qui fait partie de l'initiative COIN-OR, et KNITRO de Byrd *et al.* (2006), un solveur commercial.

On débute notre analyse du problème général (2.7), dans lequel on a introduit les variables d'écart $s$ :

$$\begin{aligned} \min_{x,s} \quad & f(x) \\ \text{s.c.} \quad & g(x) - s = 0 \quad : z, \\ & h(x) = 0 \qquad : y, \\ & s \geq 0 \qquad\quad : w. \end{aligned} \tag{E.1}$$

Notez que nous réécrivons les inégalités et égalités $c_i(x)$ sous la forme $g(x) : \mathbb{R}^n \to \mathbb{R}^m$ et $h(x) : \mathbb{R}^n \to \mathbb{R}^q$ respectivement. Elles sont sujettes aux mêmes hypothèses de continuité et différentiabilité que les $c_i(x)$. Les multiplicateurs de Lagrange qui leurs sont associés sont désignés respectivement par $z$ et $y$, tandis que $w$ est le multiplicateur de Lagrange associé aux bornes sur $s$. Nous introduisons le concept de *barrière logarithmique*, qui est essentiellement une fonction de pénalisation qui a pour but d'assurer la positivité des variables d'écart $s$. Afin de faciliter l'analyse, nous posons des variables d'écart sur toutes les inégalités, mais, en pratique, nous pouvons conserver les bornes initialement présentes dans (2.7) et leur appliquer également une barrière logarithmique. On peut donc formuler le problème *barrière* :

$$\begin{aligned} \min_{x,s} \quad & f(x) - \mu \sum_i \ln(s_i) \\ \text{s.c.} \quad & g(x) - s = 0 \qquad : \lambda, \\ & h(x) = 0 \qquad : \zeta, \end{aligned} \tag{E.2}$$

où $\mu > 0$ est le *paramètre barrière*. L'ajout d'un terme de pénalisation de forme logarithmique sur les variables d'écart permet de traiter les contraintes de bornes et $\mu$ sert à moduler son importance par rapport à $f(x)$. Typiquement, il est exigé que $\mu$ tende vers 0 au fil des

itérations afin de permettre aux solutions situées sur la frontière d'être atteintes.

Une observation des conditions de $\mathcal{KKT}$ du problème (E.2) nous permet d'en apprendre plus sur l'effet de l'ajout d'une barrière logarithmique :

$$\nabla_x \mathcal{L}(x, s; y, z) = \nabla f(x) - J_h y - J_g z = 0, \tag{E.3}$$
$$\nabla_s \mathcal{L}(x, s; y, z) = -\mu S^{-1} e + z = 0, \tag{E.4}$$
$$h(x) = 0, \tag{E.5}$$
$$g(x) - s = 0, \tag{E.6}$$
$$\tag{E.7}$$

où $e$ est un vecteur de 1, $S = \operatorname{diag}(s)$ et

$$J_h = \begin{bmatrix} \nabla h_1(x) & \nabla h_2(x) & \dots & \nabla h_q(x) \end{bmatrix},$$
$$J_g = \begin{bmatrix} \nabla g_1(x) & \nabla g_2(x) & \dots & \nabla g_m(x) \end{bmatrix}$$

sont les matrices Jacobiennes de $h(x)$ et $g(x)$ respectivement.

Par observation de (E.4), on peut conclure que le traitement de la borne $s \geq 0$ par l'ajout de la barrière logarithmique revient en fait à perturber par $\mu e$ la condition de complémentarité (2.16) des conditions de $\mathcal{KKT}$ du problème (E.1). Sans perdre de généralité, on peut réécrire (E.4) sous la forme :

$$Sz = \mu e. \tag{E.8}$$

On cherche donc un $x$ tel que les conditions de $\mathcal{KKT}$ perturbées précédentes sont respectées. Pour ce faire, on définit le système :

$$F(x, s; y, z; \mu) = \begin{bmatrix} \nabla f(x) - J_h y - J_g z \\ Sz - \mu e \\ h(x) \\ g(x) - s \end{bmatrix} = 0,$$

auquel on applique la méthode de Newton :

$$\nabla_{xsyz} F(x, s; y, z;\ \mu)\Delta = -F(x, s; y, z; \mu), \tag{E.9}$$

afin de déterminer un vecteur de pas $\Delta$ qui résoud le système. En détaillant (E.9), on obtient

le système *primal-dual* :

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & -J_h & -J_g \\ 0 & Z & 0 & S \\ J_h^\intercal & 0 & 0 & 0 \\ J_g^\intercal & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta s \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - J_h y - J_g z \\ Sz - \mu e \\ h(x) \\ g(x) - s \end{bmatrix}. \tag{E.10}$$

Dans le contexte d'un algorithme, on procède à la résolution de ce système itérativement en mettant à jour le paramètre de barrière $\mu$. On pose donc $x = x_k$ dans (E.10) et on calcule le nouveau point $x_{k+1}$ en vertu du pas $\Delta$ selon :

$$\begin{bmatrix} x_{k+1} \\ s_{k+1} \\ y_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ s_k \\ y_k \\ z_k \end{bmatrix} + \begin{bmatrix} \alpha_s^{\max}\Delta x \\ \alpha_s^{\max}\Delta s \\ \alpha_z^{\max}\Delta y \\ \alpha_z^{\max}\Delta z \end{bmatrix}. \tag{E.11}$$

On utilise les deux longueurs de pas $\alpha_s^{\max} > 0$ et $\alpha_z^{\max} > 0$ afin d'incrémenter respectivement les variables primales et duales. En pratique, il est courant de choisir un $\alpha$ tel que $s_{k+1}$ et $z_{k+1}$ n'approchent pas leurs bornes inférieures trop rapidement. L'optimalité d'un itéré $(x_k, s_k, y_k, z_k)$ est mesurée grâce au *résidu* des conditions de $\mathcal{KKT}$ (E.3), (E.5), (E.6) et (E.8) :

$$E(x, s; y, z; \mu) = \max\left\{ \left\|\nabla f(x) - J_h y - J_g z\right\|, \ \left\|Sz - \mu e\right\|, \ \left\|h(x)\right\|, \ \left\|g(x) - s\right\| \right\}. \tag{E.12}$$

On peut formaliser un algorithme de points intérieurs primal-dual générique selon l'algorithme 12. Il est important de noter que nous présentons un cadre général dans lequel on

**Algorithme 12** Méthode de points intérieurs primale-duale non linéaire

1: Étant donné $x_0$, $s_0 > 0$, calculer les mutliplicateurs $y_0$ et $z_0 > 0$. Choisir $\mu_0$ et $\sigma \ \in ]0, \ 1[$.

2: **tant que** convergence non atteinte pour (E.1) **faire**
3: **tant que** $E(x_k, \ s_k; \ y_k, \ z_k; \ \mu) > \mu$ **faire**
4: Obtenir $\Delta_k$ en résolvant (E.10)
5: Calculer $\alpha_s^{\max}$ et $\alpha_z^{\max}$
6: Calculer $x_{k+1}$ selon (E.11)
7: **fin tant que**
8: $\mu \ \leftarrow \in ]0, \ \sigma\mu[$
9: **fin tant que**

assume une fonction objectif non linéaire et des contraintes non linéaires. Dans notre cas, nous pourrions procéder à plusieurs simplifications, car nos contraintes sont linéaires et notre

objectif est convexe, de sorte que le système (E.10) aurait une forme plus simple.